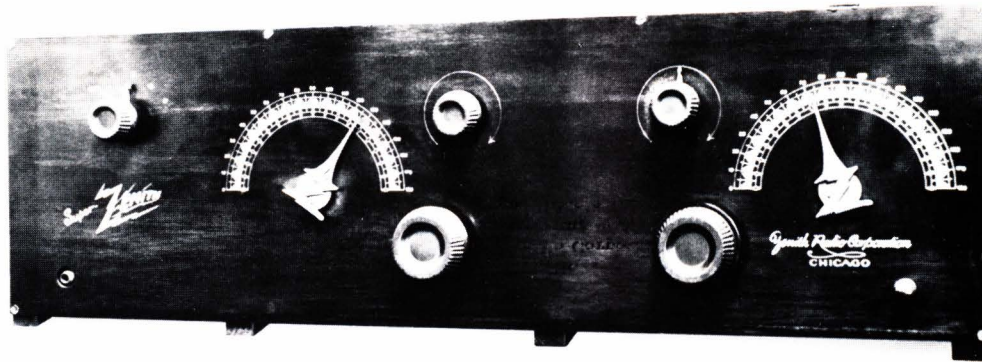


OS/2 für jedermann

am Beispiel der
Zenith-Computersysteme

- Die großen Möglichkeiten des neuen Betriebssystems
- Preisgünstiges Multitasking
- So läuft MS-DOS unter OS/2
- 90% der Befehle sind identisch
- Die Vorteile der PC/AT-Hardware ausschöpfen
- Praktische Anwendungen
- Der Einsatz im Netz
- Mit OS/2 in die Zukunft

Super Zenith,
erstes Radio von 1918



So sah vor 70 Jahren das erste Zenith-Gerät aus, nachdem 1918 Ralph Matthews und Karl Hassel in Chicago die Zenith Radio Corporation gegründet hatten. Es war ein Radioempfänger, der für damalige Verhältnisse so leistungsfähig war, daß die Produktion bis 1924 auf ein Gerät pro Tag gesteigert werden konnte.

Seitdem hat Zenith seine Rolle als Hersteller leistungsfähiger elektronischer Geräte nicht wieder aufgegeben. Immer wieder ist es den Zenith-Ingenieuren gelungen, bahnbrechende Erfindungen zu entwickeln, die modernste Technologie für den Anwender nicht nur nutzbar, sondern auch erschwinglich machte.

- 1926** Zenith erfindet das erste Radio, das am Wechselstrom-Netz betrieben werden kann.
- 1927** Zenith erfindet die Stations-Taste, mit der ein Sender auf Knopfdruck eingestellt werden kann.
- 1934** Zenith erfindet die Einknopf-Abstimmung, mittels derer durch Drehen eines einzigen Knopfes die Sender gewählt werden und wie sie bis heute üblich ist.
- 1940** Zenith nimmt die erste kommerzielle UKW-Station der Welt in Betrieb und ermöglicht damit den Durchbruch der Frequenzmodulation als Übertragungsverfahren.
- 1941** Zenith baut das erste Kofferradio, das tragbar und batteriebetrieben ist. Dieses Gerät wurde als wichtigstes Kommunikationsmittel zur Polarexpedition von Donald B. MacMillan gewählt.
- 1948** Zenith stellt dem Markt das erste Fernsehgerät vor; ein Schwarzweiß-Gerät mit einer runden Bildröhre.
- 1950** Zenith erfindet die Infrarot-Fernbedienung für Fernsehgeräte.
- 1960** Zenith ist größter amerikanischer Hersteller von Fernsehgeräten. Jeder dritte amerikanische Fernseher ist ein Zenith.
- 1971** Zenith erfindet ein völlig neues Verfahren, um die Farbdarstellung an Farbbildröhren (Beamer) erheblich zu verbessern.
- 1979** Zenith gründet den Geschäftsbereich Zenith data systems und stellt seinen ersten 8-bit Microcomputer vor, den Z-89.
- 1981** Zenith entwickelt den Z-110, einen 8- und 16-bit Computer, der zu den damaligen Standards höchste Kompatibilität bietet.
- 1983** Zenith ist der erste namhafte Hersteller der Welt, der einen vollkompatiblen PC herstellt, den Z-151 und den Portable Z-161.
- 1985** Zenith stellt seinen Z-248 dem Markt vor, den ersten 80286-PC mit Null-Wartezyklen-Technologie.
- 1986** Zenith erfindet das Supertwisted-LCD, wie es bis heute in den Portables Z-180 eingebaut ist und bisher in Größe und Darstellungsqualität noch von keinem anderen Hersteller gezeigt wird.
- 1987** Zenith ist erster Hersteller der Welt, der einen vollkompatiblen PC mit 80386 Prozessor und echtem 32-bit Bussystem baut. Eine Leistung, die ebenfalls bis heute unerreicht blieb.
- 1988** Zenith beginnt die Massenproduktion des FTM-Monitors, eines Farbbildschirmes, der vorne völlig flach ist und aufgrund neuer Technologien bisherige Farbmonitore hinsichtlich Kontrast und Helligkeit um fast das Doppelte überbietet.

Auch jetzt nach 70 Jahren ist es unser Ziel, neueste Forschungsergebnisse der Elektronik in bezahlbaren Anwendernutzen umzusetzen. Über 37000 Mitarbeiter in 30 Produktionsstätten rund um die Welt arbeiten an diesem Ziel. Denn Erfolg ist schließlich eine Geschichte, die nicht von ungefähr kommt.

Wenn Sie wissen wollen, welchen Nutzen Ihnen die technologisch führenden PC's von Zenith bringen, dann schreiben Sie uns oder rufen Sie an. Wir senden Ihnen gern einen Übersichtsprospekt und nennen Ihnen die Adresse eines Zenith-Fachhändlers ganz in Ihrer Nähe, der Sie über Einsatzmöglichkeit und Anwendungen gern ausführlich berät.



Zenith Z-386 mit FTM-Monitor, der leistungsstärkste unter den Zenith PC's

ZENITH | **data systems**

Die 100% Computer

Zenith data systems GmbH, Robert-Bosch-Str. 32-38
6072 Dreieich-Sprendlingen
Telefon: 0 61 03/39 05-0, Telefax: 0 61 03/31 931, Telex: 417986

Umsteigen im Nu

"Was bringt es, wenn ich umsteige von MS-DOS auf MS-OS/2? Wo liegen die Vor-, gibt es Nachteile? Kann ich meine Software weiter einsetzen? Muß ich mich umstellen?"

Auf all diese und viele andere Fragen, die sich beim "Prüfen wer sich ewig bindet" stellen, gibt Ihnen die vorliegende erste OS/2-Ausgabe in der CHIP-SPECIAL-Reihe Antwort.

Nur wer die Unterschiede zwischen diesen beiden Microsoft-Betriebssystemen kennt, die von IBM im deutschsprachigen Raum PC-DOS und BS/2 getauft wurden, wird ziemlich schnell zu einem Ergebnis kommen.

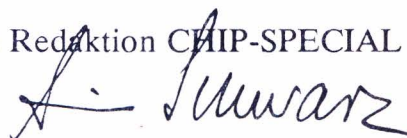
Nachdem die Auswahl an OS/2-fähiger Hardware groß ist, wir jedoch jede unserer Aussagen am praktischen Beispiel erproben wollten, haben wir uns in dieser Ausgabe auf die Produktpalette eines Soft- und eines Hardware-Herstellers beschränkt.

Obwohl alle OS/2-Software im Prinzip auf allen OS/2-Rechnern läuft, gibt es in manchen Punkten doch Unterschiede. Unterschiede, die im Zweifelsfalle die Frage nach Roß und Reiter laut werden lassen. In unserem Falle ist es im Bereich Software das Haus Microsoft, und im Bereich Hardware die deutsche Tochtergesellschaft des amerikanischen Herstellers Zenith.

Die wohl wichtigste Neuerung von OS/2 gegenüber MS-DOS ist das Multitasking. Ohne diese quasi gleichzeitige Bearbeitung von mehreren Programmen ist der Benutzer dazu verdammt, während des Ausdrucks oder des Ablaufs von Prozessen im Hintergrund, untätig vor dem Rechner Däumchen zu drehen.

Was alles im neuen Operating System steckt, vermitteln Ihnen die nachfolgenden Seiten. Dr. Norbert Meder, der als CHIP-SPECIAL-Autor die Federführung für die vorliegende Ausgabe inne hatte, meinte dazu bei der Heftablieferung aufatmend: "Im Grund ist es höchste Zeit, daß endlich ein Betriebssystem verfügbar ist, das die Möglichkeiten des 80286 ausnutzt und zudem noch die Verwendung der bereits vorhandenen MS-DOS-Software unterstützt!"

Ihre Redaktion CHIP-SPECIAL



Armin Schwarz

CHIP SPECIAL

1. Auflage 1988

Best.-Nr. 0982

Redaktionsdirektor: Richard Kerler

Chefredakteur: Armin Schwarz
(verantwortlich für den Inhalt)

Stellvertretender Chefredakteur: Ulrich Kern

Schlußredaktion: Ingeborg Kurz

Redaktionssekretariat: Petra Eibicht

Leserservice: Ingeborg Kurz

Autor dieser Ausgabe: Dr. Norbert Meder

Titelgestaltung: Hans Kuh

Titelfoto: Ezio Geneletti

Layout: Karlheinz Dereser

Redaktion: Vogel-Verlag KG Würzburg,
Redaktion CHIP-Special, Schillerstr. 23 a,
D-8000 München 2, Telefon (089) 51 49 30,
Telekopierer 53 50 00, Teletex (17) 897 190

Verlag: Vogel-Verlag KG, Postfach 67 40,
D-8700 Würzburg 1, Tel. (0931) 4 18-0,
Telex 68 883, Telefax (0931) 4 40 53,
Telegramme: CHIP-Würzburg

Verlagsdirektor: Dr. Andreas Kaiser

Anzeigenleiter: Friedrich Mangold, Würzburg
(verantwortlich für Anzeigen)

Anzeigenservice: CHIP, Postfach 67 40,
8700 Würzburg 1, Tel. (0931) 4 18-0,
Telex 68 883, Michael Belgrad,
Durchwahl 4 18-24 33.

Vertriebsleitung: Axel Herbschleb, Würzburg

Vertrieb Handelsauflage: Vereinigte Motor-
Verlage GmbH & Co. KG, Leuschnerstr. 1,
D-7000 Stuttgart 1, Tel. (07 11) 20 43-1

Bezugsmöglichkeiten: Bestellungen nehmen der
Verlag und alle Buchhandlungen im In- und Ausland
entgegen. Sollte die Zeitschrift aus Gründen, die nicht
vom Verlag zu vertreten sind, nicht geliefert werden

können, besteht kein Anspruch auf Nachlieferung oder
Erstattung vorausbezahlter Bezugsgelder.

Bankverbindungen Vogel-Verlag:

Dresdner Bank AG, Würzburg
(BLZ 790 800 52) 3 14 88 90 000,
Bay. Vereinsbank AG, Würzburg
(BLZ 790 200 76) 2 50 61 73,
Kreissparkasse Würzburg
(BLZ 790 501 30) 1 74 00,
Postscheckkonto Nürnberg
(BLZ 760 100 85) 99 91-8 53
Ausland: Postscheckkonto Zürich
80 47 064,
Niederlande 2 66 2 395
Banque Veuve Morin-Pons, Paris
155 410 314

Gesamtherstellung und Versand: VOGEL-
DRUCK WÜRZBURG, Max-Planck-Str. 7/9,
D-8700 Würzburg

Unverlangte Manuskripte werden nur zurückgesandt,
wenn Rückporto beigelegt ist. Für die mit Namen oder
Signatur des Verfassers gekennzeichneten Beiträge
übernimmt die Redaktion lediglich die presserechtliche
Verantwortung.

Die in dieser Zeitschrift veröffentlichten Beiträge sind
urheberrechtlich geschützt. Übersetzung, Nachdruck,
Vervielfältigung sowie Speicherung in Datenverarbei-
tungsanlagen nur mit ausdrücklicher Genehmigung
des Verlages.

Jede im Bereich eines gewerblichen Unternehmens
hergestellte oder benutzte Kopie dient gewerblichen
Zwecken gem. § 54 (2) UrhG und verpflichtet zur
Gebührenerstattung an die VG Wort, Abteilung Wissen-
schaft, Goethestraße 49, 8000 München 2, von der die
Zahlungsmodalitäten zu erfragen sind.

Die Redaktion hat die Manuskripte und Programme
sorgfältig geprüft. Für Fehler im Text, in Schaltbildern,
Aufbauskizzen, Listings usw. sowie deren Folgen kann
keine Haftung übernommen werden. Sämtliche Veröf-
fentlichungen erfolgen ohne Berücksichtigung eines
eventuellen Patentschutzes, auch werden Waren-
namen ohne Gewährleistung einer freien Verwendung
benutzt.

 **VOGEL** Würzburg

CIP-Titelaufnahme der Deutschen Bibliothek

Meder, Norbert:

OS/2 [OS zwei] für jedermann am Beispiel der
Zenith-Computersysteme/[N. Meder]. – 1. Aufl.
– Würzburg: Vogel, 1988

(Chip special)

ISBN 3-8023-0982-0

NE: HST

Inhaltsverzeichnis

Neue Möglichkeiten	7	1 Die großen Möglichkeiten des neuen Betriebssystems
Kompatibilität	10	2 MS-OS/2 verarbeitet MS-DOS-Dateien und Programme
Mehrfachprozesse	12	3 MS-OS/2 bietet ein preisgünstiges Multitasking
Features	14	4 Die Vorteile von MS-OS/2 an praktischen Anwenderbeispielen
Hardware-Basis	16	5 Die notwendigen Hardware-Voraussetzungen für MS-OS/2
Vorteile	18	6 Die Vorteile der Hardware ausschöpfen, aufgezeigt am Beispiel der Zenith-Computersysteme
Installation	24	7 Wie das Betriebssystem MS-OS/2 auf dem Rechner eingerichtet ist
	31	8 Die nachträgliche Installation von MS-OS/2 auf dem Rechner
Bedienung	33	9 Von MS-DOS in vier Stunden nach MS-OS/2
	36	10 Die Benutzeroberfläche: der Programm-Selektor und Session-Manager
Befehlsübersicht	42	11 MS-DOS und MS-OS/2 sind einander sehr ähnlich: 90 % der Befehle sind identisch
Neue Befehle	45	12 Die Unterschiede bei den Betriebssystembefehlen: Welche neuen Befehle sind zu beachten?
OS/2 fähige Software	49	13 So läuft MS-DOS-Software unter MS-OS/2: Im Real-Mode und verändert im Protected-Mode
Praktisch Arbeiten	51	14 Zehn Befehle genügen für den Anwender: Wer jetzt mit MS-OS/2 beginnt, tut sich ebenso leicht wie mit MS-DOS
Anwendungen für Fortgeschrittene	64	15 Befehlsmodifikationen im Protected-Mode und weitere zwölf Befehle für den fortgeschrittenen Bediener.
Desktop	75	16 MS-OS/2-Presentation-Manager: Die grafische Bedienoberfläche analog zu Windows
Netzwerk	81	17 Die Netzwerkfähigkeit von MS-OS/2: Horizontale und vertikale Vernetzung
Ausblick	85	18 Mit MS-OS/2 in die Zukunft
Editorial	3	Umsteigen im Nu
In diesem Heft	6	Das sollten Sie wissen
	4	Impressum
	91	Bestellkarten

Das sollten Sie wissen

Der Einstieg in das neue Betriebssystem MS-OS/2 richtet sich an zwei Leserkreise. Beim ersten Leserkreis gehe ich davon aus, daß er MS-DOS jedenfalls von der Bedieneroberfläche her kennt. Diese Kenntnisse müssen gar nicht sehr tief gehen. Für den Leser aus diesem Zirkel ist es fast gleichgültig, wo er in diesem Heft zu lesen beginnt. Es ist so aufgebaut, daß man fast bei jedem Abschnitt einsteigen und von dort aus in die anderen Abschnitte verzweigen kann.

Den Lesern aus dem zweiten Leserkreis, bei denen ich davon ausgehe, daß sie keine großen Kenntnisse über Personal-Computer und Betriebssysteme haben, möchte ich folgenden Zugang empfehlen: Als Überblick mögen sie den 1. Abschnitt lesen, sodann, um in die generelle Bedienung des MS-OS/2-Personal-Computers einzusteigen, den 10. Abschnitt und dann den 14. Abschnitt. Die weitere Reihenfolge ist gleichgültig. Der Abschnitt 14 enthält eine Einführung in die zehn wichtigsten Betriebssystembefehle, die keinerlei Vorkenntnisse voraussetzt.

An dieser Stelle möchte ich vor allem Christian Wedell, Geschäftsführer der Microsoft GmbH Deutschland, danken für die Überarbeitung und Ausformulierung eines ersten Rohkonzeptes, das ich für den 18. Abschnitt gemacht hatte. Wer könnte besser über das Thema "Mit MS-OS/2 in die Zukunft" schreiben als Herr Wedell?

An zweiter Stelle danke ich Peter Scheuber, der mir auch diesmal wieder eine große Hilfe bei der Verfassung dieses Manuskriptes gewesen ist.

Nun bleibt nur noch dem Leser zu sagen oder besser zuzurufen: Steigen Sie ein in die dritte Microcomputer-Generation - in die zweite Personal-Computer-Generation - in die Generation, die von MS-OS/2 angeführt wird.

Norbert Meder



1 Die großen Möglichkeiten des neuen Betriebssystems

Um mit der Tür ins Haus zu fallen: Die großen Möglichkeiten von MS-DOS/2, dem neuen Betriebssystem für Personal-Computer von Microsoft, liegen im Multitasking, in der Verwaltung größerer Speicher, in der Fähigkeit zur Datenkommunikation und in der grafischen Oberfläche. Es sei hier kurz erläutert, verständlich aber wird es erst im kompletten Heft.

Zuerst sollte man sich klar machen, was ein Betriebssystem überhaupt ist. Zur Erklärung ein Stück Computer-Geschichte: Die ersten Computer bestanden nur aus Hardware. Die Abläufe und Prozesse waren wie Einbahnstraßen oder besser wie ein Straßennetz fest zementiert und damit ein für alle Mal festgelegt. Das ist möglich, weil sich das Rechnen und die Logik in Schaltungen darstellen läßt - der sogenannten Schalt-Algebra. Im Hardware-Stadium der Computer mußten Programm- oder besser: Prozeßänderungen an der Hardware "mit dem Lötkolben" vorgenommen werden.

Dann wurde irgendwann die Programmierung erfunden. Die Idee zu dieser Erfindung liegt darin, die technische Verbindung von Schaltnetzen durch eine Art Schalt-Plan zu ersetzen, der dem Computer als Text (Code) mitgeteilt wird und der erst dann technisch realisiert wird, wenn er im Computer gelesen und sinngemäß abgearbeitet wird. Mit der Programmierung ist somit die zweite Seite der modernen Informationsverarbeitung entstanden: die Seite der Programme, die Seite der Software. Das ist die weiche (soft) Ware im Gegensatz zu den harten (hard) physikalischen Teilen des Computers.

Programme liegen in logischer Form und als Texte vor. Sie beschreiben in Form von Befehlen, Kommandos (commands), Abläufe über besondere Daten und Adressen, die der Prozessor bereitstellt. Sie beschreiben, wie die Daten des Benutzers im Speicher abgelegt, dort aufbewahrt, von dort auch wieder geholt und auch wieder dorthin zurückgebracht werden können.

Es ist mittlerweile schon kein Spezialwissen mehr, aus welchen sogenannten Systemelementen eine Com-

puter-Anlage besteht. Es sind dies der Prozessor, früher Rechen- und Steuerwerk genannt, Haupt- oder Arbeitsspeicher, Input/Output-Kanäle und die an diese Kanäle angeschlossenen Peripheriegeräte, wie z.B. Disketten-Laufwerke, Festplatte, Bildschirm, Tastatur und Drucker, Modem und vieles mehr. Wir wissen auch, daß die Konfiguration der Systemelemente den Aufbau eines Computers bestimmt. Aber wie die Abläufe der maschineninternen Arbeitsprozesse stattfinden, das weiß man zumeist nicht. Und der Benutzer oder Bediener einer Computer-Anlage, insbesondere eines Personal-Computers muß dies auch nicht wissen. Denn alle dies Abläufe steuert ein Programmpaket: das Betriebssystem. Mit dem Aufbau einer PC-Anlage sind zwar die Rahmenbedingungen vorgegeben, aber was in einem solchen Rahmen gemacht wird, das entscheidet sich am Betriebssystem.

Den Kern bildet der Prozessor, das zentrale Teil am Personal-Computer, das arbeitet und daher von der Hardware-Seite die Grenzen des Möglichen absteckt. Er erfüllt die Aufgaben einer zentralen Kommando- und Steuerstelle. Das Betriebssystem veranlaßt diese Zentrale zu den kompliziertesten und vielseitigsten Leistungen. Wie funktioniert so eine zentrale Regelung der Prozesse?

Was man mit einem Computer machen kann, ist natürlich außerdem durch die weitere Technik festgelegt: die Technik von Adreß- und Datenbus, von diversen Schnittstellen (Interfaces), von Hauptspeicher und den externen Speichern (Festplatte und Diskettenstationen). Denn diese weiteren technischen Teile verrichten ja auch handgreifliche Arbeit.

Programme real abzuarbeiten, liegt auf Seiten der Hardware. Wir sprechen in diesem Zusammenhang von den Hardware-Eigenschaften. Die Hardware-Eigenschaften der jeweiligen Prozessoren haben stets die Entwicklung von Betriebssystemen beeinflußt.

Betriebssysteme sind die Programme, die der Hardware am nächsten stehen. Sie stellen einen besonderen Teil der Software eines Systems dar. Die

Aufgabe der Betriebssystem-Programme ist nicht die Daten-Bearbeitung, sondern die Verwaltung von Programmen und Programmfunktionen, die ihrerseits Daten-Bearbeitung betreiben. Das macht Betriebssysteme zu Super-Programmen. Weil alle sogenannten Anwenderprogramme, wie z.B. Textverarbeitung, Tabellenkalkulation und anderes mehr, vom Betriebssystem abhängen und dementsprechend sinnvoll unterstützt werden müssen, braucht es von Seiten des Betriebssystems besondere Hilfsdienste, die den Benutzer in seiner Arbeit unterstützen. Wir werden diese Hilfsdienste später noch genauer kennenlernen.

Die Vorgeschichte

Erst die Mikros und dann der PC haben eine Entwicklung gefördert, in der die Rolle der Betriebssysteme gerade in Bezug auf die Hilfsdienste für den Benutzer an Bedeutung gewonnen hat. Auf den Großrechenanlagen hat es niemals so komfortable bedienerfreundliche Betriebssysteme gegeben. Das hängt nicht zuletzt daran, daß am Großrechner immer nur Spezialisten saßen. Als die Hardware für Kleinrechner billiger wurde, die Mikros an Verbreitung gewannen, da wuchsen auch die Ansprüche an die Software generell und insbesondere an ein gutes Betriebssystem. Das führte zum Konzept der Standardisierung der Software. Globale, funktionsreiche, an den Schnittstellen normierte Betriebssysteme präsentieren sich heute innerhalb gewisser Grenzen, die aber zumeist nur durch den Prozessor gesteckt sind, als weitgehend hardware-unabhängig. Sie können im optimalen Fall beibehalten werden, wenn man die Hardware wechselt.

CP/M (Control Program for Microprocessors) war wohl das erste leistungsstarke Betriebssystem, das in diesem Sinne relativ hardware-unabhängig war - eine Eigenschaft, die auch heute für Großrechner noch nicht selbstverständlich ist. CP/M war das Betriebssystem von 8-Bit-Rechnern, der alten Generation von Mikro-Computern. Es läuft auf dem Prozessor Z80 von Zilog bzw. auf dem Prozessor 8080 von INTEL. CP/M bildete lange Zeit den Standard - sowohl für die Software-Entwickler als auch für die Bediener. Microsoft-Multiplan und Microsoft-BASIC wurden noch für dieses Betriebssystem geschrieben. Es verwaltete einen Arbeitsspeicher von 64 Kilobyte - später 128 Kilobyte -, 8-Zoll oder 5,25-Zoll-Diskettenlaufwerke und in der Regel Festplatten von 10 Megabyte.

MS-DOS hat die Stellung von C/PM für die neueren 16-Bit-Rechner eingenommen. Als IBM den Personal Computer (PC) entwickelte, fragte man bei Microsoft an, ob Microsoft bereit wäre, für den PC ein Betriebssystem zu schreiben. Microsoft hatte zu der Zeit nur ein Betriebssystem in der Produktpalette: XENIX, eine lizenzierte Version von UNIX der Firma AT&T. XENIX war ein Betriebssystem für leistungsfähigere Prozessoren, als dies der 8086/8088 Prozessor von INTEL war, der dem PC zugrundeliegen sollte. So mußte ein neues Betriebssystem entwickelt werden: Microsoft-DOS. Schon bei diesem Entwicklungsprozeß spielte die Einhaltung von Standards im Hinblick auf die höchstmögliche Kompatibilität der vorhandenen Software eine entscheidende Rolle. Kompatibilität bedeutet die Verträglichkeit von Standard-Software-Programmen, wie der Textverarbeitung, mit unterschiedlicher Hardware.

So war klar, daß MS-DOS soweit wie möglich dem CP/M angepaßt sein sollte, um dem 16-Bit-Rechner von IBM den Markt zu öffnen. Es ergab sich sogar noch passender. Weil IBM in Eile war, ihren Rechner anzubieten, kaufte Microsoft das Betriebssystem SCP-DOS von Seattle Computer Products, das für einen CP/M-80-"Clone" geschrieben war und auf dem 8088-Prozessor lief, und benutzte dieses Produkt als Ausgangsbasis für MS-DOS.

Als MS-DOS im August 1981 in der Version 1.0 herauskam, war es nur für den IBM-PC bestimmt. Die Version 1.1 erschien 1982 und arbeitet mit zweiseitigen 320 Kilobyte Floppy-Diskettenlaufwerken. Das Ziel war erreicht: MS-DOS war noch mehr kompatibel zu CP/M-80 als das Betriebssystem aus dem Mutterhaus von CP/M - als CP/M-86. CP/M-Applikationen wie Microsoft-Multiplan konnten leicht portiert werden. Die Orientierung am alten CP/M-Standard hat Kontinuität in der technologischen Entwicklung von Mikro-Computer-Systemen gewahrt. Das war sicherlich gut so - kostengünstig und bedienerfreundlich. Es brachte aber auch Beschränkungen mit sich, die für diesen Preis zu zahlen waren. Viele Schwächen von MS-DOS hätten vermieden werden können, wenn es nicht um den damals bestehenden Standard gegangen wäre.

Das neue Betriebssystem MS-OS/2

Nun stehen wir vor der Verbreitung eines neuen Betriebssystems. Wenn man so will, dann steht jetzt die dritte Generation eines Mikro-Computer-Standards

an. Diesmal sind die Voraussetzungen günstiger, die Entwicklung durchdachter, weil auch die Entwicklungszeit länger war. Das hervorgebrachte Produkt ist gelungen. MS-OS/2 bewahrt auf raffinierte Weise einerseits die bis heute geschaffenen Standards und setzt andererseits neue Standards für die Zukunft.

Denn MS-OS/2 ist ein doppeltes Betriebssystem: Es präsentiert in einem Teil MS-DOS und wahrt die Tradition, im anderen Teil bietet es den Multitasking-Betrieb und zeigt in die Richtung unserer zukünftigen Arbeit. Denn Multitasking heißt, daß mehrere Programme gleichzeitig auf ein und demselben Rechner laufen und bedient werden können. Das bedeutet weniger Wartezeiten, höhere Leistung und höhere Effizienz.

Der englische Ausdruck für Betriebssysteme ist OPERATING SYSTEM. Von daher kommen die beiden Buchstaben OS im Namen von so vielen Betriebssystemen vor. Sehr bekannte PC-Betriebssysteme heißen einfach DOS (disk operating system). Der Ausdruck DOS signalisiert auch, daß für eine ganze Generation von Betriebssystemen die Disketten-Operationen das dominante Problem waren. Das hat sich jetzt mit MS-OS/2 geändert. Diese Disketten-Operationen gehören heute zum Standard. Wenn das neue Betriebssystem von Microsoft nicht mehr DOS, sondern OPERATING SYSTEM (OS) heißt, dann deutet dies zurecht an, daß damit die Kinderzeit der Mikros endgültig vorbei ist.

Mit Multitasking wechselt das dominante Thema bei Mikros von Disketten-Operation zu Programm-Operation (komplexe Prozeß-Steuerung). MS-OS/2 ist das Betriebssystem mit den zwei Gesichtern, dem alten MS-DOS-Gesicht und dem neuen Gesicht, das viele Aufgaben gleichzeitig sieht (Multitasking).

Mehr Speicher

Der entscheidende Faktor für MS-OS/2 auf der Hardware-Seite sind die Prozessoren 80286 und 80386 von INTEL - das sind Prozessoren, die im IBM AT, in den PS/2-Systemen von IBM, in den Zenith-Rechnern Z-248, Z-286, Z-386, in den Portablen der Turbo-Serie sowie in den vielen anderen kompatiblen Rechnern eingebaut sind. Diese Prozessoren können vor allem ein Vielfaches von Speicher verwalten im Vergleich zu ihrem Vorgänger, dem 8088/8086-INTEL-Prozessor. MS-OS/2 greift auf diese Möglichkeit zu.

MS-OS/2 baut auf ihnen auf. Aber es ist vermutlich besser ausgedrückt, wenn wir sagen, daß MS-OS/2 die Fähigkeiten des 80286 und des 80386 zu allererst voll zugänglich macht. Denn das Betriebssystem ist die wesentliche Seite dessen, was man mit einem Computer machen kann. Erst durch ein geeignetes Betriebssystem wird Produktivität freigelegt. Produktivität beruht wesentlich auf dem breiten Angebot von Software, die auf einem Standard im Bereich der Betriebssysteme aufbaut. Diesen Standard hat MS-OS/2 neu gesetzt.

Zwei Betriebsarten

Der 80286 von INTEL ist mit einem Janus-Kopf vergleichbar. Er hat zwei Gesichter. Das eine kennen die meisten von Ihnen. Es ist die Betriebsart, in der auch bisher der 80286 im AT oder Kompatiblen unter dem Betriebssystem MS-DOS im Gebrauch ist. Daß der 80286 noch ein zweite Seite hat, ist vielen bislang unbekannt geblieben. Dieses zweite Gesicht steht für eine Betriebsart, die den 80286 gerade von seinen Vorgängern 8086 und 8088 unterscheidet und die jetzt unter MS-OS/2 den Multitasking-Betrieb ermöglicht. Daß der Anwender nun beide Betriebsarten - im beliebigen Wechsel - wählen und in Gebrauch nehmen kann, das ist die Leistung des neuen Betriebssystems. MS-OS/2 dreht für Sie den Janus-Kopf, damit Sie stets in das Gesicht sehen können, das Ihnen gefällig ist.

MS-OS/2 löst die hochkomplizierten und hochkomplexen Steuerungsaufgaben, die bei einem solchen zweiseitigen Prozessor anfallen. Aber das ist nur die Aufgabe zur Hardware-Seite hin. Zur anderen Seite hin gilt es, dem Anwender - gleichgültig, ob er Programmierer oder Software-Bediener ist - den Zugriff zu allen Diensten zu verschaffen, die der 80286 bietet. MS-OS/2 ist nämlich als zentrale Software das Mittelglied zwischen Hardware und Bediener.

Grafische Bedieneroberfläche

Als ein solcher Vermittler bietet MS-OS/2 in der Form des Presentation-Managers, der sich genauso wie Windows 2.0 bzw. wie Windows 386 präsentiert, eine grafische Benutzeroberfläche an. Auch wenn dieser Presentation-Manager zur Zeit noch nicht fertiggestellt ist, so weiß man doch, wie er aussehen wird. Denn seine beiden Brüder sind bekannt. Und daß mit der Windows-Oberfläche ein zukünftiger

Standard gesetzt wird, belegt die Einigung von IBM und Microsoft über diesen Problempunkt. Was hat es auf sich mit der Grafik und den Betriebssystemen?

Bei weitem die schnellste Weise, Informationen aufzunehmen, geschieht durch das Bild und nicht durch das Wort oder den Befehl. Und die bei weitem schnellste Weise, Informationen zu geben, ist das Zeigen. Dies zu berücksichtigen wird umso wichtiger, je komplexer die Systeme werden. Denn dies verlangt auch vom Bediener die Erfassung und Handhabung hochkomplexer Informationen. Wenn MS-OS/2 im Multitasking nebeneinander und gleichzeitig ausführt, was der Benutzer in parallelen Aufgaben zu lösen wünscht, dann ist auch der Bediener in diesem hochkomplexen Geschäft zu unterstützen. Er braucht in Zukunft nur noch mit der Microsoft-Maus in lexikalischen (wörtlichen) oder piktografischen (bildhaften) Menüs auf das zu zeigen, was er machen will und schon wird die Funktion ausgeführt.

Er wird in Fenstern, die sich überlappen und hintereinander liegen können, die verschiedenen Anwendungen ansehen können, die das Betriebssystem und die leistungsfähigen Prozessoren zu bieten haben. Damit ist auch schon die weitere Aufgabe angesprochen, die jedes Betriebssystem von diesem Leistungsumfang erfüllen muß: die Interprozeß-Kommunikation. Sie stellt sich von der Benutzer- oder Anwender-Seite als die Notwendigkeit dar, Daten zwischen verschiedenen Anwendungen von einem Fenster in das andere Fenster verschieben zu können.

2 MS-OS/2 verarbeitet MS-DOS-Dateien und Programme

Die kleine Geschichte des Personal-Computer zeigte schon, wie wichtig Standards sind und wie ernst die Fragen nach Verträglichkeit (Kompatibilität) und nach dem Zusammenhang des Alten mit dem Neuen (Kontinuität) in der Bedienung genommen werden. Die technologische Entwicklung ist derart rasant, daß sie droht, den Menschen in seiner Fähigkeit, der Entwicklung zu folgen und an ihr aktiv teilzunehmen, zu überrollen. Das kann vermieden werden, wenn Kompatibilität und Kontinuität gewahrt und Standards gesetzt werden. Wie MS-OS/2 dies macht, wird im Folgenden kurz umrissen.

Dynamischer Datenaustausch und Schutzvorrichtungen

Bei allen Abläufen von Programmen bietet MS-OS/2 die Möglichkeit zum sogenannten dynamischen Datenaustausch, das heißt: wenn sich beispielsweise in der Tabellenkalkulation, die in einem Fenster des Betriebssystems abläuft, gewisse Zahlen verändern, dann zeigt sich dies zugleich in einem zweiten Fenster an der entsprechenden Grafik. Das Betriebssystem übernimmt die Steuerung der dazu notwendigen internen, in der Hardware verankerten Prozesse. MS-OS/2 verfügt über Steuerungskonzepte, nach denen die Anwenderprogramme abgearbeitet werden und gleichzeitig miteinander kommunizieren, das heißt Daten austauschen. MS-OS/2 tut dies auf einer noch höheren Ebene, als man dies bislang für Personal-Computer gewohnt war: im Multitasking!

Und insbesondere für den Multitasking-Betrieb sind spezielle Hilfs- und Schutz-Einrichtungen nötig. Denn die parallel laufenden Programme dürfen mit ihren unterschiedlichen Daten nicht in ein chaotisches Durcheinander geraten. Das heißt unter anderem, daß kein Programm ohne die Erlaubnis des Betriebssystems direkt auf die Hardware zugreifen darf. MS-OS/2 löst diese Probleme, indem es Dienstleistungen zur Verfügung stellt, wie sie der Benutzer von Mikros bislang nicht gekannt hat. Auf die ausgeklügelten Steuerungsmechanismen werden wir später noch zu sprechen kommen.

Aber vorab schon das Wichtigste: MS-OS/2 ist zu 99 % mit MS-DOS kompatibel. MS-OS/2 arbeitet mit allen DOS-Daten-Dateien und mit fast allen MS-DOS Programmen. Wo es Schwierigkeiten gibt, sind wenige Anpassungen notwendig, von denen man ausgehen kann, daß die großen Software-Hersteller sie rasch vollzogen haben.

Einige Hintergrundinformationen sind wichtig, um zu verstehen, wie MS-OS/2 dies geschafft hat. Als INTEL den Prozessor 80286 und später 80386 auf den Markt brachte, da waren von Seiten der Hardwa-

re die Voraussetzungen für MS-OS/2 geschaffen. Der 80286 präsentiert sich als Prozessor für zwei Betriebsarten, der 80386 sogar für vier Betriebsarten.

Der Real-Mode

Eine Betriebsart des 80286 ist der Real-Mode. Er stellt weitgehend die Art und Weise dar, wie die Prozessoren 8088 und 8086 arbeiten. Die Leistungsgrenzen dieser Prozessoren sind Ihnen vielleicht bekannt: Trotz raffinierter Speicherverwaltung (Adressierungstechnik) kann nur maximal 1 Megabyte Arbeitsspeicher zur Verfügung gestellt werden. In der Regel sind es sogar nur 640 Kilobyte. Das hängt hauptsächlich damit zusammen, daß diese beiden Prozessoren (8088 und 8086) nur physikalische Adressierung kennen. Eine Speicheradresse muß wirklich (real) auch einem Hardware-Speicher im Hauptspeicherbereich entsprechen. Deshalb heißt dieser Arbeitsmodus auch Real-Mode.

Die Adressierung von Speichern im Computer ist von zentraler Bedeutung, weil nur über interne Adressen Speicherinhalte von Daten und Programmen gefunden werden können. Den Adreßraum stellt der Prozessor zur Verfügung, das Betriebssystem nutzt ihn aus. Nur mit großen Speicher- und Adreßräumen ist das zu verwirklichen: Multitasking, Daten- und Prozeß-Kommunikation und grafische Verarbeitung.

Ist der Adreßraum real, entspricht ihm also wirklicher Speicher auf der Hardware-Seite, dann muß ein Programm auch fest (statisch) im Arbeitsspeicher bleiben, solange es läuft. Diese feste Bindung an die physikalische Basis läßt dem Betriebssystem kaum eine Chance, steuernd einzugreifen. Das ist nicht schlimm, solange immer nur ein Programm-Prozeß abläuft. In Konflikte gerät man erst, wenn es notwendig oder wünschenswert wird, daß mehrere Prozesse gleichzeitig ablaufen. Dann gibt es unter anderem Konkurrenz um den Speicherplatz - sowohl im Hinblick auf die Programme als auch im Hinblick auf den Platz, den die Programme für die bearbeiteten Daten benötigen. Wenn solche Verhältnisse eintreten, wird ein umfassender Steuerungsmechanismus vom Betriebssystem erforderlich.

Man kann nicht sagen, daß eine solche Steuerung im Real-Mode absolut unmöglich ist. Aber der Aufwand von Seiten der Programmierung stünde in keinem vernünftigen Verhältnis zum Ertrag. Die Prozessoren

8088 und 8086 sowie der Real-Mode des 80286 bieten dazu einfach zu wenig Unterstützung. Das ist in der zweiten Betriebsart des 80286, dem Protected-Mode, anders. Aber darauf kommen wir später zurück.

Das Wichtigste hier an dieser Stelle ist die Tatsache, daß mit dem Real-Mode von Seiten des Prozessors die Möglichkeit gegeben ist, ganz normal wie bisher - nur schneller, weil der Prozessor einen höheren Zeittakt zuläßt - unter MS-DOS zu arbeiten. Das ist aus der Bedienung von AT's und den Kompatiblen, wie z.B. dem Zenith Z-248 und dem Zenith Z-286, bekannt. Diese Möglichkeit zu bewahren, in einem schnellen 8086-Mode arbeiten zu können, war eine der großen Aufgaben, die MS-OS/2 zu erledigen hatte. Und das war beim 80286 gar nicht so einfach.

Real und Protected-Mode: das Umschalten

Das Ergebnis kann vorweggenommen werden: Dem Prozessor 80286 entsprechend kennt auch MS-OS/2 zwei Zustände - den Real-Mode und den Protected-Mode. Man kann MS-OS/2 in zwei Betriebsarten verwenden. Eigentlich besitzen Sie mit MS-OS/2 zwei Computer: einen Computer, der im wesentlichen unter dem Betriebssystem MS-DOS von der Version 3.1, 3.2 oder 3.3 arbeitet und einen Multitasking-Computer, der unter dem ganz neuen Teil von OS/2 arbeitet.

Die Betriebsart des Real-Mode ist mit MS-DOS ab Version 3.x kompatibel - das "x" steht hier für irgendeine Zusatznummer zur Version - oder besser gesagt: Die alten unter MS-DOS geschriebenen Programme sind mit dieser Betriebsart verträglich. Deshalb wird dieser Zustand Ihres Betriebssystems auch Compatibility-Mode genannt.

Damit hat sich MS-OS/2 optimal an den Prozessor 80286 angepaßt. Aber wie schon gesagt, dies zu verwirklichen war wohl das härteste Stück Arbeit in der Entwicklung von OS/2. Denn der 80286 hat eine schwerwiegende Schwäche. Er läßt sich während des Betriebs nur in eine Richtung umschalten - vom Real-Mode zum Protected-Mode. Umgekehrt vom Protected-Mode zum Real-Mode geht das nicht. Damit sind im Prozessor die beiden Betriebsarten eigentlich nicht miteinander verträglich.

Lange Zeit hat man es auch in der OS/2-Entwicklergruppe von Microsoft und IBM für unmöglich gehalten, die Verträglichkeit der beiden Betriebsarten - sozusagen von der Software-Seite kommend - herzustellen. Tatsächlich hatte man es nach anfänglichen Bemühungen schon aufgegeben, als dem Chef-Designer von MS-OS/2, Gordon Letvin, die mittlerweile patentierte Idee kam, den Zugriff auf beide Betriebsarten dadurch zu ermöglichen, daß der Prozessor beim Umschalten kurzfristig ganz abgeschaltet wird. Das ist in Amerika vielfach mit der Tatsache verglichen worden, die Zündung eines PKW bei 90 km/h aus- und einzuschalten.

Der Prozessor 80286 startet (bootet) im Real-Mode. Vom Real-Mode in den Protected-Mode umzuschalten, ist auch vom Betriebssystem her kein Problem. Will man allerdings zurückschalten, dann muß sozusagen der Prozessor ganz abgeschaltet und neu gestartet werden. Dieses Verfahren, Kompatibilität herzustellen, kommt einem Geniestreich gleich. Zumal da in diesem Ab- und Umschaltverfahren der Zustand des Arbeitsspeichers, das heißt laufende Prozesse und Daten, gerettet und bewahrt werden müssen. Daß dies alles gelungen ist, macht nun aus, daß der Bediener mit dem neuen Betriebssystem MS-OS/2 genauso wie bisher arbeiten kann.

Im Real-Mode können Sie also wie bisher Ihre geläufige Software verwenden, z.B. Microsoft-Word, Microsoft-Multiplan, Microsoft-Chart, Microsoft-Rbase

und viele andere Software-Produkte auch. Also keine Angst vor dem neuen Betriebssystem!

Darüber hinaus präsentierte Zenith mit den ersten Rechnern, die überhaupt mit dem Betriebssystem MS-OS/2 ausgeliefert wurden, eine weitere Möglichkeit, mit der bisherigen Software weiterzuarbeiten. Zenith arbeitete mit Microsoft bei der Entwicklung von MS-OS/2 eng zusammen. Das ermöglichte Zenith eine Startroutine in ihre Rechner einzubauen, die den Bediener fragt, ob er unter MS-DOS oder unter MS-OS/2 booten will. Man kann sich auf diese Weise entscheiden, ob man für eine Arbeitssitzung an seinem PC unter dem einen oder dem anderen Betriebssystem arbeiten möchte. Allerdings ist bei diesem Verfahren ein Umschalten während der Arbeitssitzung nicht möglich. Um diese Abfrage neu zu erhalten, muß man den Rechner neu booten.

Zenith war der erste Hersteller, noch vor IBM und anderen, der im Dezember 1987 einen PC anbot, der erstens standardmäßig MS-DOS als Betriebssystem enthielt und für einen Aufpreis darüber hinaus als zweites neues Betriebssystem MS-OS/2. Damit bot Zenith als erster Hersteller einen preisgünstigen Multitasking-Rechner auf dem Markt an. Was alles mit Multitasking in Verbindung steht, soll im folgenden Abschnitt erläutert werden. Dabei wird auch jene zweite Betriebsart des Prozessors 80286 genauer beschrieben.

3 MS-OS/2 bietet ein preisgünstiges Multitasking

Der Protected-Mode sieht bestimmte Steuerungs- und Schutzmechanismen für den Multitasking-Betrieb vor. Im wesentlichen ist das die sogenannte virtuelle Speicherverwaltung. Virtuell ist das lateinische Fremdwort für "möglich", wörtlich heißt es "der Kraft nach". Hier genügt es, wenn Sie sich klarmachen, daß der von einem Programm benötigte Speicherplatz zuerst einmal nur in logischen Adressen beschrieben wird. Bildhaft gesprochen, steht das alles erst nur auf dem Papier. In der "Realität" kann das dann ganz anders und auch jeweils verschieden aussehen. So etwas ist jedem aus dem Alltag bekannt. Die Realität ist hier die Hardware, das heißt der wirklich vorhandene Arbeitsspeicher. Die mögliche (virtuelle)

Speicherbelegung steht natürlich hier nicht auf dem Papier, sondern in Beschreibungstabellen, die schon der Prozessor 80286 im Protected-Mode bereitstellt. Logische und physikalische Adressierung werden dadurch miteinander in Einklang gebracht, daß neben dem Arbeitsspeicher der Speicherplatz der Festplatte mit in die Speicherplatzverwaltung einbezogen wird. Das ist kein Problem, weil Sie in aller Regel auf der Festplatte genügend Platz haben. Im Einzelnen geschieht das so:

Der virtuelle Speicher für Ihr - sagen wir - Statistikprogramm ist größer als der Raum, den der Prozessor real im Arbeitsspeicher zur Verfügung stellen

kann. Also bekommt Ihr Programm nur das, was im Arbeitsspeicher da ist, und der Rest muß auf der Festplatte bleiben. Der virtuelle oder logische Speicher wird auf die beiden physikalischen bzw. realen Speicher, Arbeitsspeicher und Festplattenspeicher, verteilt.

Gehen wir jetzt davon aus, daß mehrere Programme parallel im Hauptspeicher abgearbeitet werden sollen. Der Effekt dieser ausgeklügelten Speicherverwaltung ist dann unter anderem, daß jedem Programm nur ein verhältnismäßig kleiner Platz physikalisch im Arbeitsspeicher zur Verfügung gestellt werden muß. Der Rest kann ja auf der Festplatte bleiben. Das macht die Konkurrenz um den Raum im Hauptspeicher überflüssig. Jedem der konkurrierenden Programme wird einfach ein sogenannter privater Bereich zuerkannt. Und ein solcher Bereich ist dann gegen alle anderen geschützt (protected). Dieser Umstand hat dann auch der Betriebsart den Namen Protected-Mode gegeben.

Trotz dieser raffinierten Speicherplatz-Verwaltung käme man natürlich mit den bisher - beim 8088/8086 - üblichen Speichern von 640 Kilobyte nicht sehr weit. Denn im heutigen Durchschnitt dürften die meisten Programme schon für sich genommen um die 300 Kilobyte einnehmen. Darüber hinaus bearbeiten sie ja auch noch Daten, die im Hauptspeicher gehalten werden müssen. Gibt man solchen Programmen zu wenig realen Speicher, dann muß zuviel zwischen Hauptspeicher und Festplatte hin und her bewegt werden. Im Englischen nennt man dies Swapping, was auch der Fachausdruck bei Rechnern geworden ist.

Um also zuviel Swapping zu vermeiden, hat man den Prozessor 80286 im Protected-Mode so angelegt, daß bei der Speicherplatz-Verwaltung die erweiterte Nutzung von bis zu real 16 Megabyte im Hauptspeicher und 1 Gigabyte auf der Festplatte möglich ist. Notwendig sind allerdings nur real 1,5 Megabyte im Arbeitsspeicher - empfehlenswert 4 Megabyte.

Wenn man bedenkt, daß man diese erweiterten Arbeitsmöglichkeiten jetzt schon weitgehend bezahlt hat, wenn man z.B. einen Z-286 von Zenith kauft, und daß es nur des Aufpreises von weniger als 600 Mark bedarf, um diese Möglichkeiten auch mit MS-OS/2 nutzen zu können, dann hat man ein preisgünstiges Multitasking. Außerdem zeigt sich schon jetzt in einem so frühen Stadium der Entwicklung, daß auch die Standard-Software unter MS-OS/2 nicht teurer sein wird als bisher. Microsoft kündigte seine Tabellenkalkulation Multiplan in zwei neuen Versionen 4.0 an - eine DOS-Version und eine OS/2-Version - beide zum jeweils gleichen Preis. Dasselbe gilt für die Datenbank Microsoft-Rbase-System.

Die wirtschaftliche Auslastung der Rechenzeit war von Anfang an ein Hauptproblem der EDV. Die Prozessoren sind meistens so schnell, daß sie fast niemals zu 100% ausgelastet sind. Denn immer dann, wenn von Programmen zum Beispiel Daten zum Bildschirm, zum Drucker, auf die Festplatte geschickt oder von der Platte geholt werden, entstehen Leerlaufzeiten. Diese Daten-Transport-Prozesse dauern verhältnismäßig lang, und der Prozessor muß warten.

Im Multitasking wird die bessere Auslastung des Rechners nach folgendem Muster erreicht: Das Betriebssystem kontrolliert, welche und wieviele Aufgaben von der Tastatur aus, das heißt letztlich vom Bediener, angemeldet sind. Jede Aufgabe stellt einen Programmablauf dar, dem innerhalb des Arbeitsspeichers ein eigener Teilbereich zugeordnet wird. Von Zeit zu Zeit - in unmerklich kleinen Zeitintervallen - fragt das Betriebssystem jeden dieser Teile ab, ob das dortige Programm etwas abzuarbeiten hat. Wenn dies der Fall ist, startet das Betriebssystem die entsprechenden Prozesse. Nach Ablauf eines festgelegten Zeitintervalls auf einer sogenannten Zeitscheibe wird das Programm wieder unterbrochen, damit das Betriebssystem zur nächsten Aufgabe übergehen kann. Auf diese Weise wird die Prozeß-Zeit so aufgeteilt, daß jede Task (jedes einzelnes Anwendungsprogramm) der Reihe nach ein Stück in seinem Prozeß weitergeführt und abgearbeitet wird.

4 Die Vorteile von MS-OS/2 an praktischen Anwenderbeispielen

Im Protected-Mode können also Programme im Multitasking abgearbeitet werden, das heißt viele Aufgaben können annähernd gleichzeitig gelöst werden. Das bedeutet ja Multitasking. Multi heißt "viel" und task ist die zu erledigende "Aufgabe".

Multitasking heißt also, daß von einem Benutzer mehrere Programme gleichzeitig in den Hauptspeicher geladen werden. Das Betriebssystem verteilt die Rechenzeit annähernd gleichmäßig und in festem Rhythmus an alle anstehenden Programme und beginnt mit der Bearbeitung des ersten. Noch bevor dieses erste Programm vollständig zu Ende gekommen ist, beginnt das Betriebssystem mit der Bearbeitung des zweiten Programmes und so weiter.

Drucken und Schreiben in der Textverarbeitung

Eine einfache Anwendungsmöglichkeit: Die Sekretärin möchte einen längeren Text ausdrucken. Normalerweise ist dann der Bildschirm für lange Zeit blockiert, denn Drucken braucht Zeit. Im Multitasking-Betrieb aber kann sie parallel zum Drucken gleichzeitig an einem anderen Text weiter-schreiben. Das Drucken ist in den Hintergrund verlegt.

Sortieren und Daten erfassen in einer Datenbank

Oder nehmen Sie etwa die Datenerfassung in einem Datenbankprogramm, die Sie gleichzeitig vornehmen wollen, während irgendwelche Datenbank-Tabellen im Hintergrund neu sortiert werden sollen, dann geht das letztlich nur im Multitasking. Das Betriebssystem regelt im Vordergrund, das heißt am Bildschirm, die Eingabe der Daten. Währenddessen wird das Sortierprogramm im Hintergrund gleichzeitig abgearbeitet.

Kalkulation in zwei verschiedenen Tabellen

Soweit es aus den Ankündigungen zu Microsoft-Multiplan für OS/2 hervorgeht, bietet diese Multitasking-Version folgende Arbeitsmöglichkeiten: Der Anwender kann mehrere Kalkulationen, das heißt mehrere Rechentabellen, die in verschiedenen OS/2-Dateien liegen, gleichzeitig bearbeiten. Langwierige Berechnungen, wie z.B. Konsolidierungen der Daten, können unbemerkt im Hintergrund von statten gehen, während der Anwender im Vordergrund, das heißt am Bildschirm, kalkulatorische Planspiele zu anderen Daten durchführt. So ist der Rechner jederzeit optimal auszulasten. Gleichzeitig - wir werden noch sehen, wie das geht - kann der Bediener stets von einem laufenden Prozeß zum anderen umschalten (switchen) und nachsehen, wie weit es mit der Programmarbeit steht oder welches Ergebnis schon vorliegt.

Dateneingabe und Rekalkulation in einer Tabelle

Multitasking erlaubt aber auch, daß bei der Tabellenkalkulation gleichzeitig zur Dateneingabe eine Rekalkulation in der gerade bearbeiteten Tabelle abläuft, das heißt in ein und derselben OS/2-Datei. Grundsätzlich kann in OS/2 solch ein Vorgang über Prioritäten derart gesteuert werden, daß die Neuberechnungen im sichtbaren Ausschnitt der Tabelle vorgezogen werden. Im sichtbaren Ausschnitt der Tabelle heißt, in dem Bereich, der auf dem Bildschirm zu sehen ist. Dabei kann sogar eine Verschiebung der Tabellen auf dem Bildschirm berücksichtigt werden.

Inneres und äußeres Multitasking

An den beiden Beispielen zur Tabellenkalkulation kann man auch sehr schön den Unterschied zwischen zwei Formen von Multitasking sehen. Das erste Beispiel betrifft ein äußeres Multitasking, das heißt ein Tabellenkalkulations-Programm kann gleichzeitig auf zwei verschiedenen DOS- bzw. OS/2-Dateien arbei-

ten. Die Datenbereiche sind also getrennt. Das zweite Beispiel steht für inneres Multitasking. Das heißt das Programm kann sich intern in verschiedene Prozesse aufteilen, die parallel laufen, zur Dateneingabe einerseits und Rekalkulation andererseits. Ein sehr gutes Beispiel für dieses programminterne Multitasking ist auch das oben genannte Beispiel aus der Textverarbeitung.

Rechtschreibprüfung beim Text-Schreiben

In der Textverarbeitung gibt es noch bessere Fälle für den Einsatz von parallel laufenden, intern unterschiedenen Prozessen. Heute ist in der Textverarbeitungs-Software zumeist ein Modul für Rechtschreibprüfung eingebaut. Man ruft solche Rechtschreibungs-Module nach der Texteingabe auf und läßt sie ablaufen. Multitasking macht es möglich, solche Rechtschreibprüfungen parallel ablaufen zu lassen und dies sogar auf zwei Weisen. Einmal in der Weise, daß das System einen fertig gestellten Text prüft, während wir gleichzeitig einen anderen Text schon wieder eingeben oder aber auf eine zweite Weise, die noch sehr viel komfortabler ist, nämlich dadurch, daß das System schon während der Texteingabe im Hintergrund die Rechtschreibung prüft und jeden Fehler sofort meldet. In diesem letzteren Fall wären Texteingabe und Rechtschreibprüfung auf denselben Text bezogen und vollständig ineinander verschränkt.

Multitasking und der Datenbankbereich

Unserer eigenen Einschätzung nach wird einer der bedeutendsten Einsätze von MS-OS/2 im Bereich von Datenbank Anwendungen liegen. Denn erstens geht es bei ausgefeilten Datenbanken immer um Massendaten, das heißt zugleich: es muß in lang dauernden Prozessen viel gearbeitet werden. Und zweitens gibt es kaum einen Bereich, in dem die Datenpflege so wichtig ist wie bei Datenbanken. Denn Datenfehler können hier unüberschaubare Folgen haben. So ist es unter MS-OS/2 durchaus denkbar, daß sich die Datenbanksoftware in die Richtung entwickelt, daß Datensicherungs-routinen, Plausibilitätsprüfungen und Veränderungsprotokolle im Multitasking parallel abgewickelt werden. Gerade das Führen von Veränderungsprotokollen ist für eine professionelle Datenbank-anwendung unerlässlich. Denn nur über solche

Protokolle können bei fehlerhafter Eingabe Daten im Nachhinein wieder rekonstruiert werden.

Ein letztes Beispiel soll aus dem Bereich der Datenkommunikation gewählt werden, gleichgültig ob Sie in einem lokalen Netz einer Arbeitsgruppe angeschlossen sind oder an das öffentliche Netz, und in diesem Fall zum Beispiel per Modem oder ähnlichen Einrichtungen mit anderen Teilnehmern des öffentlichen Netzes Daten tauschen wollen. In jedem Fall handelt es sich beim Datenaustausch um Prozesse, die ohne Multitasking für längere Ablaufzeiten Ihren Rechner blockieren würden. Mit MS-OS/2 ist es kein Problem, solche Austauschprozesse im Hintergrund neben der sonstigen Arbeit ablaufen zu lassen.

Grundsätzliches zum Multitasking

Die Beispiele sind so ausgewählt, daß an ihnen sichtbar wird, wie MS-OS/2 arbeitet. Im Vordergrund (am Bildschirm) läuft das eine Programm benutzerorientiert ab. In allen "Pausen", die zwangsläufig dadurch entstehen, daß der Benutzer zum Beispiel bei der Eingabe im Verhältnis zum Rechner zu langsam ist oder auch dadurch, daß der Rechner intern nur Daten transportiert, aber nicht rechnet, in all diesen Pausen werden nun die anderen Programme im Hintergrund abgearbeitet. Grundsätzlich wird die Verarbeitung der vielen Prozesse beim Multitasking folgendermaßen gesteuert: Alle Prozesse, die für die Verarbeitung bereitstehen, kommen ohne Unterschied in eine sogenannte Warteschlange. Über eine Zeitscheibe (Round Robin) wird die Verarbeitungszeit auf die einzelnen Prozesse annähernd gleichmäßig verteilt. Da der Prozessor im allgemeinen nicht ausgelastet ist und die Befehlseingaben des Benutzers im Vordergrund in der Regel gepuffert werden, merkt der Benutzer von dieser intramaschinellen Arbeitsteilung so gut wie gar nichts.

Die Steuerung von "vielen Aufgaben", die gleichzeitig gestellt sind, verlangt eine spezielle intramaschinelle Organisation. Die vielen Aufgaben werden nämlich nicht wirklich gleichzeitig gelöst. Alle Computer bis zur sogenannten heutigen "vierten Generation" sind nach dem ungarisch-deutschen Mathematiker benannte Von-Neumann-Maschinen, die sequentiell (der Reihe nach) und nicht parallel arbeiten. Das heißt, es gibt keine wirklich gleichzeitigen Prozesse. Und wenn dies trotzdem so scheint, dann doch nur, weil der Bediener vom Zeitscheiben-Konzept nichts merkt und weil die Prozeßgeschwindigkeit heute so

enorm hoch ist, daß unser menschliches Wahrnehmungsvermögen hinsichtlich der kleinen Zeitunterschiede versagt.

MS-OS/2 löst also die Zeitverteilung mit dem Zeitscheiben-Konzept. Das Betriebssystem teilt jedem in einem gewissen Zeitrahmen angemeldeten Programm der Reihe nach eine gewisse Zeitscheibe zu. In dieser Zeiteinheit wird es abgearbeitet. Dann kommt das nächste Programm an die Reihe und so weiter. Dabei berücksichtigt MS-OS/2 auch vorgegebene Prioritäten. So entsteht im Ganzen eine Quasi-Gleichzeitigkeit im Echtzeitbetrieb oder besser: im Direktzeitbetrieb.

Das bedingt auch eine kompliziertere Regelung der Prozesse im Hinblick auf die Umgebungen, in denen die einzelnen Prozesse ablaufen können. Umgebungen - das sind vor allem die Informationen über die Pfade und Verzeichnisse auf den Datenträgern sowie über die benötigten Betriebsmittel (wo befinden sich die Daten, wohin soll die Ausgabe erfolgen usw.), die nun komplexer zu verwalten sind. Außerdem müssen - wie schon erwähnt - Angaben zur Priorität von Programmen (eilige Programme müssen vorgezogen werden) und Ähnlichem gemacht werden. Dem einzelnen Programm selbst kann der Zugriff auf den

Prozessor nun nicht mehr überlassen werden. Könnte jedes Programm direkt Einfluß nehmen, dann würde es im Multitasking-Betrieb fast zwangsläufig zu Kollisionen kommen.

MS-OS/2 regelt die Steuerung mehrerer Programme durch ein Verfahren, das man Time-Sharing nennt. Die Prozessorzeit wird auf verschiedene Prozesse aufgeteilt. Protected-Mode heißt der Zustand Ihres Rechners in einem solchen Verfahren, weil in ihm einerseits der Prozessor auf besondere Weise geschützt ist: In die Regelung des arbeitsteiligen Ablaufs vieler Prozeduren darf und kann nicht mehr direkt eingegriffen werden. Auf der anderen Seite müssen auch die Arbeitsbereiche der einzelnen Programme geschützt (protected) werden. Das leistet die schon erläuterte virtuelle Speicher-Verwaltung.

Man sieht an diesen kurzen Bemerkungen schon leicht, daß die beiden Konzepte, das der Speicherverwaltung und das der Zeitverteilung, für einen Multitasking-Betrieb - obwohl schon jedes für sich genommen komplex genug - zusammenspielen und zusammengehören. Beide Konzepte greifen im konkreten Multitasking-Betrieb ineinander. MS-OS/2 regelt deshalb auch beides zugleich: die Verteilung von Speicherplatz und Prozeßzeit.

5 Die notwendigen Hardware-Voraussetzungen für MS-OS/2

Nach den einleitenden Informationen über MS-OS/2, die Ihnen sicherlich einen ersten Überblick verschafft haben, soll es nun konkreter werden. Wenn Sie sich einen OS/2-Rechner - z.B. von Zenith - kaufen, dann erhalten Sie natürlich eine lauffähige Hardware-Konfiguration, so daß Sie sofort zu arbeiten anfangen können. Sie werden aber dennoch wissen wollen, welche Hardware-Ausstattung Sie mindestens brauchen und welche empfehlenswert ist. Zur Verwendung des Betriebssystems MS-OS/2 brauchen Sie:

Notwendig

einen 16-Bit-Personal-Computer

mit mindestens 2 Megabyte RAM Hauptspeicher, 1,5 Megabyte für den OS/2 Multitasking-Betrieb im Pro-

TECTED-Mode und 0,5 Megabyte für die MS-DOS-Anwendungen im Real-Mode

mit einem Prozessor vom Typ 80286 oder 80386 von INTEL

mit einem 1,2 Megabyte Diskettenlaufwerk und einer Festplatte von 20 Megabyte Speicherplatz

mit einer Video-Karte, die mindestens den monochromen (schwarz/weiß) Herkules-Standard in der grafischen Auflösung liefert

Theoretisch könnte Ihr Computer mit einer kleineren Festplatte ausgestattet sein, aber dies ist heute nicht mehr gängig und auch nicht zu empfehlen, weil es kaum praktikabel ist. Das 1,2-Megabyte-Laufwerk braucht man, weil Teile von MS-OS/2 größer sind als

360 Kilobyte und deshalb nicht auf die kleineren Disketten passen. Und man muß stets die Möglichkeit behalten, auch von der Diskette booten (starten) zu können. Es ist sicherlich damit zu rechnen, daß als Alternative auch die 3,5-Zoll-Laufwerke mit 720 Kilobyte oder 1,44 Megabyte in Frage kommen.

Eine empfehlenswerte Ausstattung:

3 Megabyte RAM Hauptspeicher, 1 Megabyte für den Real-Mode und 2 Megabyte für den Protected-Mode

30/40 Megabyte externer Speicher auf der Festplatte

eine Video-Karte, die den EGA-Farbgrafik-Standard liefert

ein Monitor, der gute Grafik-Auflösung bringt

Sieht man für den Real-Mode 1 Megabyte RAM Hauptspeicher vor, dann sind auch in der Kompatibilitätsbox die vollen 640 Kilobyte, die direkt unterstützt werden können, vorhanden. Die vollen 640 Kilobyte in der Kompatibilitätsbox von OS/2 zur Verfügung zu haben, ist deshalb so wichtig, weil die Betriebsart im Real-Mode von MS-OS/2 nicht exakt dieselbe ist, wie die des alten MS-DOS. Während bei MS-DOS nur ca. 60 Kilobyte im Arbeitsspeicher belegt werden, sind dies schon ungefähr 100 Kilobyte im Real-Mode von OS/2. Es bleiben also für Applikationen bei dieser vorgeschlagenen Konfiguration nur 540 Byte frei. Dies ist auch eine - wenn auch nicht sehr bedeutende - Kompatibilitätsgrenze zu MS-DOS.

Eine Festplatte von der Größenordnung von 30/40 Megabyte sei deshalb empfohlen, weil das Betriebssystem MS-OS/2 schon allein annähernd 2 Megabyte Speicherplatz auf der Festplatte einnimmt. Hat man darüber hinaus noch MS-DOS zusätzlich installiert, was bei Zenith-Rechnern grundsätzlich möglich ist - wir werden das später sehen - dann kommen ein weiteres halbes Megabyte hinzu. Bedenkt man außerdem, daß die gegenwärtige Standard-Software nicht nur funktionsreicher, sondern auch im Speicherbedarf größer wird, dann kann man sich leicht einen Rechner-Arbeitsplatz vorstellen, der allein für Programme, die auf der Festplatte liegen, 20 Megabyte beansprucht.

Die Empfehlung für ein Farbgrafik-Display, das heißt für einen Farbgrafik-Monitor und die entspre-

chende Video-Karte, beruht auf der beobachtbaren Tendenz zur grafischen Oberfläche der gängigen Standard-Software, die durch MS-OS/2 unterstützt wird. Je komplexer die Software auf der Oberfläche wird, desto größer wird die Erleichterung für die menschliche Wahrnehmung und Strukturierung, wenn wir für notwendige visuelle Unterscheidungen Farbe hinzunehmen. Wer viel an einem Personal-Computer arbeitet, wird sehr rasch die geringere Belastung für die Konzentration und auch ganz schlicht für die Augen feststellen können. Genauso wie höhere Komplexität der Programme eine Erleichterung der Arbeit bedeuten, genauso bedeuten sie auch höhere Anforderungen an den Bediener. Diese höheren Anforderungen können durch ergonomischen Komfort gestützt, teilweise kompensiert und erleichtert werden.

Gleichgültig für welches Video-Konzept man sich entscheidet: Derzeit ist für MS-OS/2 wichtig, daß man die Einstellung für einen Grafik-Modus auf der Video-Karte entsprechend feststellt. Es mag Ihnen bekannt sein, daß es derzeit auf dem Markt Video-Karten gibt, die sich automatisch auf den jeweils besten Grafik-Modus in Abhängigkeit von Programm und Bildschirm einstellen. Dieses sogenannte Autoswitch-Verfahren wird augenblicklich von MS-OS/2 noch nicht unterstützt, weswegen es notwendig ist, die auch auf diesen Video-Karten vorhandenen Dip-Schalter z.B. fest auf EGA-Modus einzustellen.

Für den Fall, daß Sie schon im Besitz eines Zenith-Rechners sind, der die genannten Grundvoraussetzungen für den Betrieb von MS-OS/2 mitbringt, dann müssen Sie für die mögliche Umstellung auf das neue Betriebssystem noch folgendes beachten:

Jeder Rechner besitzt einen Speicherbaustein, in dem geringe Teile des Betriebssystems fest verankert sind. Dieser Baustein heißt ROM (read only memory). Er kann nur gelesen (read only), aber nicht beschrieben werden. Er enthält vor allen Dingen die Routinen, die das System hochfahren lassen. Die Zenith-ROM's, ein Eigenprodukt des Herstellers, waren schon stets zum Industriestandard hochkompatibel. Das kann natürlich nicht heißen, daß ältere ROM's auch alle zukünftigen Neuerungen schon beinhalten können. So ist es wichtig zu wissen, daß für den Betrieb von MS-OS/2 nur Zenith-ROM's von der Version 1.9 oder höher verwendbar sind. Die meisten Zenith-Rechner, die alle anderen Bedingun-

gen für MS-OS/2 mitbringen, sind mit solchen ROM's bestückt. Aber natürlich kann es auch Aus-

nahmen geben. Deshalb ist auf diesen Punkt, der ja nur die Umrüstung betrifft, besonders zu achten.

6 Die Vorteile der Hardware ausschöpfen, aufgezeigt am Beispiel der Zenith-Computersysteme

Im August 1987 lieferte die Microsoft GmbH Deutschland die erste Version von MS-OS/2, das sogenannte Software-Development-Kit (Bausatz für die Programm-Entwicklung), an die Software-Entwickler aus. Man kann natürlich davon ausgehen, daß größere Software-Hersteller schon früher mit Vor-Versionen von MS-OS/2 versorgt worden waren. Jedenfalls war der August 1987 der Monat, in dem das neue Betriebssystem in den Markt einbrach. Bei diesem Start war Zenith als erster Hersteller kompatibler PC's mit dabei. Am 31. August 1987 schloß Zenith einen Vertrag mit Microsoft zur Vermarktung des neuen Betriebssystems MS-OS/2. Diese rasche Zusammenarbeit erklärte Zenith in einer Pressemitteilung vom 9. September 1987 wie folgt:

"Dies wurde möglich, nachdem Zenith Data Systems in den letzten Monaten bei der Entwicklung von MS-OS/2 sehr eng mit Microsoft zusammengearbeitet hatte. Wesentliche Bestandteile des MS-OS/2 wie z.B. die Schnittstellen zu Video, Massenspeicher und I/O-Geräten wurden in den Labors von Zenith entwickelt."

Zenith begann schließlich mit der Auslieferung der ersten Rechner, bei denen man wahlweise MS-OS/2 oder MS-DOS bestellen konnte, im November 1987. Damit war für den Endbenutzer von Personal-Computern der erste MS-OS/2-Rechner erhältlich. Aus der Sicht des Benutzers war dieses Ereignis der eigentliche Beginn der Ära von MS-OS/2. Fast ein historisches Ereignis. Andere Hardware-Hersteller zogen mit ihren MS-OS/2-Rechnern erst später nach.

Mit dem Betriebssystem MS-OS/2 liefert Zenith Data Systems die Version 2.0 von Microsoft-Windows aus. Für den Bediener sieht Windows 2.0 so aus wie der Presentation-Manager für MS-OS/2, der in der Entwicklung noch nicht ganz fertig ist. Er wird später von Microsoft und IBM mit dem Betriebssystem direkt mitausgeliefert. Er stellt dann das Standard-

Grafik-Interface für alle Personal-Computer dar und bietet eine einheitliche Benutzer-Oberfläche, was ein großer Vorteil sowohl für die Software-Hersteller als auch für die Bediener ist.

Mit der Freigabe der ersten MS-OS/2-Rechner verstärkte Zenith seinen Anspruch auf eine führende Rolle als Hersteller von kompatiblen Personal-Computern. Gleichzeitig will das Unternehmen damit deutlich machen, "daß MS-OS/2 keineswegs nur ein Produkt ist, das einem großen Hersteller exklusiv vorbehalten bleibt". So kommentierte jedenfalls der Geschäftsführer von Zenith-Deutschland, Werner Rein-kendorff, das Vorpreschen des Herstellers im Bereich von Rechnern, die unter dem neuen Betriebssystem MS-OS/2 laufen können.

OS/2-fähige Rechner

Hier werden alle Rechner aus der Zenith-Produktpalette vorgestellt, die OS/2-fähig sind. Selbstverständlich ist jeder dieser OS/2-fähigen Rechner gleichzeitig auch uneingeschränkt MS-DOS-tauglich. Aber das Umgekehrte gilt nicht. Zenith hat natürlich auch reine MS-DOS-Rechner im Programm, die nicht OS/2-fähig sind. Diese Rechner bleiben hier unerwähnt.

Für den Betrieb unter MS-OS/2 sind einige wichtige Eckdaten zu beachten. So muß jeder OS/2-Rechner mit einem 80286-Prozessor bzw. mit einem 80386-Prozessor von INTEL ausgestattet sein. Diese Bedingung ist bei allen Modellen, die wir hier besprechen, erfüllt. Zum anderen ist für den Betrieb mit MS-OS/2 ein Arbeitsspeicher von mindestens 1,5 Megabyte erforderlich. Beim Kauf eines OS/2-fähigen Rechners sollten Sie beachten, daß die meisten der vorgestellten Rechner - ebenso wie bei anderen Bewerbern am Markt - in der Grundausstattung noch nicht über diese 1,5 Megabyte verfügen, so daß der Arbeitsspeicher auf den entsprechenden Wert aufge-

rüstet werden muß. Eine Ausnahme hiervon ist einer der tragbaren Zenith-Rechner, nämlich der TurboSPORT-386, der bereits in der Grundausstattung 2 Megabyte Arbeitsspeicher mitbringt und deshalb in dieser Grundausstattung mit MS-OS/2 betrieben werden kann. Die beiden hier vorgestellten portablen Rechner - das ist zum einen der bereits erwähnte TurboSPORT-386 und zum anderen sein kleinerer Bruder, der SupersPORT-286 - bieten einige interessante technische Details. Sie sind zum Beispiel auch für den netzunabhängigen Betrieb geeignet.

Der Zenith Z-386

In einer häufig verwendeten Unterscheidung spricht man von 8-Bit-, 16-Bit- oder 32-Bit-Rechnern. Doch diese Unterscheidung ist manchmal nicht exakt genug. Sie betrifft nämlich nur den Mikroprozessor. Auf jeden Fall ist es sinnvoll, nicht nur auf den Mikroprozessor zu achten, sondern auch auf die Breite des sogenannten Datenbusses, auf dem die Daten zwischen dem Prozessor, dem Speicher und den peripheren Einheiten ausgetauscht werden. So spricht man z.B. von einem 16-Bit-Prozessor, wenn dieser Mikroprozessor intern mit 16-Bit-Werten rechnen kann. Man muß sich dies so vorstellen: Der Prozessor verarbeitet immer geschlossen ein Wort aus 8 Buchstaben (Zeichen/Bit) oder ein Wort aus 16 Buchstaben oder schließlich ein Wort aus 32 Buchstaben. Je länger die Worte sind, mit denen er geschlossen umgehen kann, desto schneller wird die Verarbeitungsgeschwindigkeit im Ganzen. Wenn nun ein Prozessor - wie heute üblich - "16-Bit-Wörter" verarbeitet, dann bedeutet dies jedoch nicht automatisch, daß auch der Datenbus 16 Bit breit ist. Der Mikroprozessor 8088 von INTEL ist z.B. ein 16-Bit-Mikroprozessor, der allerdings nur mit einem 8 Bit breiten Datenbus arbeitet. Das hat dann zur Folge, daß zwei Speicherzugriffe erforderlich sind, um z.B. ein 16 Bit breites internes Register des Prozessors zu laden. Dagegen verfügt der Prozessor 8086, der ebenfalls ein 16-Bit-Mikroprozessor ist, gleichzeitig auch über einen 16 Bit breiten Datenbus. Dann können auch "16-Bit-Wörter" geschlossen "verschickt werden", das heißt ein 16-Bit-Register des Prozessors kann mit einem Adressierungsvorgang geladen werden.

In diesem Fall, wenn also sowohl ein 16-Bit-Prozessor als auch ein 16 Bit breiter Datenbus vorliegt, dann spricht man von einem "echten" 16-Bit-Rechner. Dementsprechend findet man auch Rechner mit einem 32-Bit-Prozessor, die aber nur

einen 16 Bit breiten Datenbus besitzen. Dann ist es ein 16/32-Bit-Rechner; in dieser Sprechweise bezieht sich die 16 auf die Breite des Datenbusses und die 32 auf die Breite der prozessor-internen Register.

Der Zenith Z-386 nun ist ein echter 32-Bit-Mikrocomputer, das heißt er verfügt über einen 32-Bit-Mikroprozessor und einen 32 Bit breiten Datenbus. Der verwendete Mikroprozessor 80386 von Intel bietet im Vergleich zu den anderen Intel-Prozessoren eine höhere Verarbeitungsgeschwindigkeit und damit einen besseren Datendurchsatz. Gleichzeitig gewährleistet der Z-386 volle Kompatibilität zum Industriestandard. Zenith bedient sich darüber hinaus einiger Technologien, die zum Teil selbst entwickelt wurden und die Leistungsfähigkeit des Z-386 beträchtlich steigern. Damit wird MS-OS/2 noch besser unterstützt. Hier einige dieser Leistungsmerkmale.

Der Mikroprozessor

Der Mikroprozessor 80386 von Intel, der wie sein Vorgänger, der 80286, über den für MS-OS/2 notwendigen Protected-Mode verfügt, wird im Z-386 mit einer Taktfrequenz von 16 MHz betrieben. Die Taktfrequenz legt die Anzahl der elementaren Arbeitsschritte fest, die ein Mikroprozessor in einer Sekunde ausführen kann. Die Höhe der Taktfrequenz, mit der ein Prozessor arbeiten kann, wird dabei begrenzt durch die Zeit, die die elektronischen Schaltkreise brauchen, um einen dieser elementaren Arbeitsschritte vollständig auszuführen. Der Z-386 bietet volle Kompatibilität zu den PC-Standards. Das bedeutet, daß Programme, die für PC's mit 8088/86 oder 80286 geschrieben in unveränderter Form auf dem Z-386 ablauffähig sind. Allerdings laufen diese Programme dann erheblich schneller ab, weil die 32-Bit-Architektur des 80386 für erheblich kürzere Befehlsausführungszeiten sorgt. Unter der Architektur eines Mikroprozessors versteht man die logisch-technische Bauweise der Schaltkreise. Aufgrund des 32-Bit-Datenbusses sind AT/PC-kompatible 32-Bit-Erweiterungen möglich. Darüber hinaus ist gleichzeitig noch Platz für PC-kompatible 8- und AT/PC-kompatible 16-Bit-Steckkarten vorhanden. Mit diesen Steckkarten kann die Hardware des Z-386 den jeweiligen Erfordernissen angepaßt werden.

Der Cache-Speicher

Für den Z-386 bietet Zenith - wenn man dies wünscht - den Einbau eines 64 Kilobyte großen Cache-Speichers an. Dieser Cache-Speicher besteht aus sehr schnellen Speicherbausteinen mit einer Zugriffszeit von nur 35 ns (eine Nanosekunde ist eine milliardstel Sekunde). Die aktuellen Daten, mit denen der Prozessor arbeitet, können im Cache-Speicher gehalten werden und sind damit schneller verfügbar. So ergeben sich insbesondere für rechenintensive Programme (z.B. CAD, DTP, Compiler, etc.) erhebliche Geschwindigkeitsvorteile, die sich vor allem im Multitasking von MS-OS/2 deutlich bemerkbar machen.

Die RAM-Speicher

Der Speicherzugriff beim Z-386 erfolgt ohne Wartezyklen. Wartezyklen sind immer dann erforderlich, wenn die Speicherbausteine langsamer arbeiten als der Prozessor, das heißt der Prozessor muß eine gewisse Anzahl von Taktzyklen warten, bis die aus dem Speicher abgerufenen Daten zur Verfügung stehen. Ein Taktzyklus ist die Zeitdauer, die dem Reziprokwert der Taktfrequenz entspricht. Bei einer Taktfrequenz von 1 MHz, also einer Million Hertz beträgt die Dauer eines Taktzyklus exakt eine Millionstel Sekunde. Bei 16 MHz ist dagegen ein Taktzyklus nur noch etwas über 80 Nanosekunden lang. Daß der Z-386 trotz seiner Taktfrequenz von 16 MHz mit 0-Wartezyklen arbeitet, liegt daran, daß Hochleistungs-RAM-Bausteine verwendet werden, die mit der hohen Geschwindigkeit des Prozessors mithalten können. Das bedeutet hier konkret, daß spätestens 80 Nanosekunden nachdem ein Speicherbaustein adressiert worden ist, die angeforderten Daten auf dem Datenbus anliegen müssen. Im Falle des Z-386 bedingt das eine Leistungssteigerung um 30 Prozent. Damit erreicht der Z-386 eine Durchsatzrate von insgesamt 4 MIPS (das sind 4 Millionen Befehle pro Sekunde) und stößt damit in den Bereich von Systemen der mittleren Datentechnik vor.

Das Enhanced Page-Mode-RAM

Mit RAM (Random Access Memory) wird der Arbeitsspeicher eines Rechners bezeichnet. Auf den RAM-Speicher eines Computers kann üblicherweise nur Zelle für Zelle zugegriffen werden. Der Zenith Z-386 verfügt jedoch darüber hinaus über die Möglich-

keit, ganze Blöcke (Speicherseiten) mit einer Größe von 2 bzw. 4 Kilobyte in einem einzigen Adressierungsvorgang zu lesen oder zu schreiben. Die Übertragung solcher Datenblöcke wird durch einen speziellen Page-Mode-Controller gesteuert. Auf diese Weise wird der Datendurchsatz um 40 Prozent erhöht.

Burst-Mode Memory Refresh

Der Inhalt der dynamischen RAM-Speicher muß innerhalb relativ kurzer Zeitabstände regelmäßig aufgefrischt werden (refresh). Bei herkömmlichen Speicherarchitekturen muß dieser Refresh-Vorgang Zelle für Zelle abgewickelt werden. Während des Refreshs ist der Systembus blockiert und kann von anderen Einheiten nicht benutzt werden. In diesem Zusammenhang ist es sehr wichtig zu wissen, daß fast die gesamte Kommunikation innerhalb eines Rechners über den Systembus abläuft. Unter anderem wird der gesamte Datentransport zwischen Prozessor und Arbeitsspeicher allein über den Systembus abgewickelt. Um den Systembus zu entlasten, werden deshalb beim Z-386 mit jedem Refresh-Zyklus gleich 4 Reihen Speicher auf einmal aufgefrischt. Dieses Vorgehen wirkt sich positiv auf die Gesamtleistung des Systems aus.

Das Slushware-RAM

Unter dem ROM (Read Only Memory) versteht man im Gegensatz zum RAM ein Speicherelement, das nur gelesen aber nicht beschrieben werden kann. In üblichen PC's ist eine Reihe von Systemfunktionen fest im ROM gespeichert (sog. ROM-BIOS-Funktionen). Diese Systemfunktionen werden beim Z-386 aus dem ROM in das Slushware RAM geladen. Bei dem Slushware RAM handelt es sich um einen sehr schnellen Speicherbereich, so daß der Zugriff auf das relativ langsame ROM unnötig wird. Dadurch wird eine beträchtliche Leistungssteigerung erzielt, da die nun im Slushware RAM abgelegten Informationen vom System häufig benötigt werden.

Die Video-Schnittstelle

Der Z-386 ist mit einer Grafikkarte nach dem VGA-Standard (640 x 480 Punkte mit 16 aus 64 möglichen Farben) ausgestattet. Gleichzeitig unterstützt die Grafikkarte alle bisher üblichen Grafikstandards (EGA, CGA, HGC). Der Z-386 paßt sich mit dieser Karte

automatisch an die jeweils erforderliche Grafikdarstellung an, so daß Probleme mit untereinander nicht kompatiblen Grafikstandards endgültig der Vergangenheit angehören. Hier zeigt sich die große Erfahrung, die Zenith aus dem Bereich der Fernseh- und Videotechnik mitbringt.

Der ROM-Monitor

Bereits im ROM der Zenith-Rechner integriert befindet sich ein sogenannter ROM-Monitor. Wenn Sie den ROM-Monitor mit der Tastenkombination <Strg-Alt-Einf> aufrufen, dann erscheint die folgende Meldung (Bild 6.1).

Wenn Sie nun ein Fragezeichen eingeben und die Enter-Taste drücken, wird Ihnen der volle Befehlsumfang des ROM-Monitors angezeigt (Bild 6.2).

Besonders interessant ist hier die Möglichkeit, daß man den Rechner von beliebigen Laufwerken bzw. beliebigen Festplattenpartitionen booten kann. Damit können verschiedene Betriebssysteme, die sich auf unterschiedlichen Partitionen befinden, auf sehr bequeme Art und Weise aktiviert werden. So gibt man, um z.B. vom Festplattenlaufwerk C: zu booten, einfach die Zeichenfolge "B W0" ein. Der ROM-Monitor ermöglicht weiterhin den direkten Zugriff auf Speicher- und Ein-/Ausgabeadressen, Register und Informationen über die momentane Konfiguration des Rechners. Außerdem stellt er einfache Debugging-Hilfen zur Verfügung.

Technische Daten

Der Zenith Z-386 besitzt in der Grundausstattung 1 Megabyte Enhanced-Page-Mode-RAM, das in Schritten von 1 bzw. 4 Megabyte bis auf 16 Megabyte aus-

```
MFM-300 Monitor, Version 2.0F
Memory Size: 640K + 4096K
Enter "?" for help.
- >
```

Bild 6.1

- MFM-300 Command Summary -		
CMD:	Explanation	Syntax
?:	Help	?
B:	Boot from disk	B [{F W}][{0 1 2 3}][[:<partition>]]
C:	Color bar	C
D:	Display memory	D [<range>]
E:	Examine memory	E <addr>
F:	Fill memory	F <range>,{<bytes>!"<string>"}
G:	Execute (Go)	G [=<addr>][,<breakpoint>]...
H:	Hex math	H <number1>,<number2>
I:	Input from port	I <port>
M:	Move memory block	M <range>,<dest>
O:	Output to port	O <port>,<value>
R:	Examine Registers	R [<register>]
S:	Search memory	S <range>,{<bytes>!"<string>"}
T:	Trace program	T [<count>]
U:	Unassemble program	U [<range>]
V:	Set Video/Scroll	V [M<mode>][S<scroll>][100][300]
	Where <range> is:	<addr>{,<addr>:L<length>}
TEST:	Extended diagnostics	TEST
SETUP:	Define hardware Setup	SETUP
	Copyright (C) 1987, by Zenith Data Systems	
->		

Bild 6.2

gebaut werden kann. Unter MS-OS/2 erscheint ein Ausbau auf 4 Megabyte sinnvoll. Ein Sockel für den mathematischen Coprozessor 80387 ist ebenfalls vorhanden. Weiterhin verfügt der Z-386 über ein Diskettenlaufwerk mit 1,2 Megabyte und eine Festplatte mit 40 oder 72,5 Megabyte. Die Zugriffszeiten auf die Festplatte liegen bei 40 Millisekunden. Darüber hinaus sind Einbauplätze für eine zweite Festplatte und für ein zweites Diskettenlaufwerk oder für einen Streamer vorhanden. Eine parallele und eine serielle Schnittstelle, sowie 7 Slots für Steckkarten, von denen 5 frei verfügbar sind, runden das Bild ab.

Der Z-386 wird standardmäßig mit MS-DOS 3.2 und MS-Windows ausgeliefert. MS-OS/2 ist auf Wunsch und gegen Aufpreis erhältlich. Außerdem wird, wie bei allen Zenith-Rechnern üblich, eine ausführliche Dokumentation mitgeliefert.

Der Zenith Z-248/12

Der Z-248/12 ist speziell im Hinblick auf OS/2 entwickelt worden. Er arbeitet mit dem INTEL-Prozessor 80286 der für OS/2 mindestens erforderlich ist. Der Prozessor wird hier mit der relativ hohen Taktfrequenz von 12 MHz betrieben und benötigt dennoch keine Wartezyklen. Der Speicherausbau beträgt 1 Megabyte in der Grundausstattung und läßt sich auf der Grundplatine (!) auf ganze 6 Megabyte aufrüsten. Das wird durch die Verwendung von SIMM's (Single Inline Memory Modules) erreicht, die äußerst platzsparend auf der Platine untergebracht werden können. Selbstverständlich läßt sich der Arbeitsspeicher mit Steckkarten auf die vollen 16 Megabyte aufstocken, die vom 80286 direkt adressiert werden können. Der Z-248/12 ist auch für die Aufnahme eines Coprozessors 80287 vorbereitet, für den bereits ein Sockel vorhanden ist. Als Betriebssysteme bietet Zenith MS-DOS, MS-OS/2 und Xenix an. Die Video-Schnittstelle bildet die schon vom Z-386 her bekannte VGA-Karte. Außerdem verfügt der Z-248/12 über eine parallele und zwei serielle Schnittstellen, sowie 7 Erweiterungssteckplätze, von denen 5 frei sind. Um für jeden Anwender eine optimale Konfiguration zu bieten, präsentiert sich der Z-248/12 in fünf Modellvarianten, die jeweils mit einer Festplatte versehen sind. Dabei handelt es sich um die Modelle 40, 40+, 80, 80+ und das Modell 160, wobei die Zahlen die jeweilige Festplattengröße in Megabyte bezeichnen. Alle fünf Modelle haben ein 3,5"-Diskettenlaufwerk mit einer Kapazität von 1,44 Megabyte. Die Modelle mit einem "+" sind zusätzlich mit einem

5,25"-Laufwerk mit 1,2 Megabyte ausgestattet. Dadurch wird der Z-248/12 unabhängig vom jeweils vorliegenden Diskettenformat und kann Daten und Programme zwischen diesen Formaten hin und her schieben, was die Übernahme von MS-DOS-Programmen auf 3,5"-Disketten ermöglicht.

Der Zenith Z-286 LP

Mit dem Z-286 LP bietet Zenith ein für den Betrieb mit OS/2 konzipiertes Einstiegermodell an. Der 80286 wird hier mit 8 MHz getaktet und arbeitet ohne Wartezyklen. Auf Wunsch ist der Z-286 LP mit dem Coprozessor 80287 lieferbar. In der Grundausstattung ist 1 Megabyte vorhanden (640 Kilobyte Arbeitsspeicher, 256 Kilobyte Expanded Memory und 128 Kilobyte ROM). Wie der Z-248/12 läßt sich der Arbeitsspeicher des Z-286 LP auf der Grundplatine bis auf 6 Megabyte erweitern. Als Video-Schnittstelle steht eine EGA-Karte mit einer Auflösung von 640 x 480 Punkten zur Verfügung, die gleichzeitig auch Software unterstützt, die für andere Grafik-Standards, wie CGA, MDA und Hercules, geschrieben ist. Der Z-286 LP ist - wie der Z-248/12 - mit einer parallelen und zwei seriellen Schnittstellen ausgestattet. Für Erweiterungen sind zwei freie 16-Bit-Steckplätze vorhanden. Vom Z-286 LP werden zwei Modelle angeboten. Beide Modelle sind mit einem 3,5"-Diskettenlaufwerk mit 1,44 Megabyte versehen. Die Modelle 20 und 40 unterscheiden sich ausstattungsmäßig lediglich in der Größe der mitgelieferten Festplatte (20 Megabyte oder 40 Megabyte).

Der Zenith Z-248 advanced PC

Beim Z-248 advanced PC handelt es sich um einen AT-kompatiblen Rechner. Der eingebaute Mikroprozessor INTEL 80286 bildet die Grundlage für den Betrieb von MS-OS/2. Er wird im Z-248 mit 8 MHz getaktet. Der Speicherzugriff erfolgt hier ebenfalls wie beim Zenith Z-386 ohne Wartezyklen. Auf Wunsch wird zum Z-248 auch der Co-Prozessor 80287 angeboten. Der Arbeitsspeicher beträgt 640 Kilobyte in der Grundausstattung. Dieser Arbeitsspeicher ist in Blöcken zu je 1,5 Megabyte auf bis auf 3,5 Megabyte erweiterbar. Als Betriebssysteme werden neben MS-DOS 3 und MS-OS/2 auch Microsoft-Xenix und andere angeboten. Als Videoschnittstelle ist ein Enhanced Graphics Adapter (EGA) erhältlich. Diese Grafikkarte paßt sich automatisch den üblichen Grafikstandards an, wie z.B. EGA, CGA,

MDA und Hercules. Dabei können 16 aus 64 Farben mit 640 x 350 Punkten dargestellt werden. Alternativ ist auch ein Monochrom-Adapter erhältlich. In der Grundausstattung wird der Z-248 mit einem 1,2 Megabyte-Diskettenlaufwerk ausgeliefert. Zusätzlich werden Festplatten mit 20 und 40 Megabyte angeboten. Ein zweites Festplattenlaufwerk ist ebenfalls anschließbar. Wie gewohnt sind eine serielle und eine parallele Schnittstelle vorhanden. Außerdem besitzt der Z-248 5 freie Erweiterungssteckplätze, davon 4 mit 8/16 Bit und einer mit 8 Bit.

Der Zenith Z-286 Compact-advanced PC

Der Z-286 Compact-advanced PC wurde bereits im Hinblick auf MS-OS/2 konstruiert. Sein 80286-Prozessor wird, wie beim Z-248, mit 8 MHz getaktet, arbeitet aber hier mit einem Wartezyklus. Der arithmetische Co-Prozessor 80287 ist ebenfalls auf Wunsch erhältlich. Der Arbeitsspeicher der Grundausstattung beträgt 640 Kilobyte und ist auf 3,1 Megabyte erweiterbar. Standardmäßig wird der Z-286 mit MS-DOS 3.2 ausgeliefert. Außerdem sind wie beim Z-248 auch Microsoft-Xenix und selbstverständlich MS-OS/2 erhältlich. Als Videoschnittstelle wird die bereits vom Z-386 bekannte VGA-Grafikkarte verwendet. Diese Karte stellt sich automatisch auf alle gängigen Bildschirmstandards ein, wie z.B. auch EGA, CGA, MDA und Hercules. Der Z-286 Compact-Advanced PC ist in drei Varianten erhältlich: Das Modell 1 wird mit zwei 3,5"-Diskettenlaufwerken mit jeweils 720 Kilobyte ausgeliefert. Das Modell 2 verfügt über ein 3,5"-Diskettenlaufwerk mit 1,44 Megabyte und ein Festplattenlaufwerk mit 20 Megabyte. Das Modell 3 wird mit einem 5,25"-Diskettenlaufwerk mit einer Kapazität von 1,2 Megabyte und einem Festplattenlaufwerk mit 20 Megabyte ausgeliefert. Außerdem verfügt der Z-286 Compact-Advanced PC über eine parallele und eine serielle Schnittstelle sowie drei freie Erweiterungssteckplätze, davon zwei mit 8/16 Bit und einer mit 8 Bit.

Die Portablen: TurbosPORT-386 und SupersPORT-286

Beim TurbosPORT-386 handelt es sich - wie auch beim TurbosPORT-286 - um einen AT-kompatiblen Rechner. Eine Neuigkeit ersten Ranges ist, daß der TurbosPORT-386 schon in der Grundausstattung nicht nur mit MS-DOS, sondern auch mit MS-OS/2

betrieben werden kann, da er über einen Arbeitsspeicher von 2 Megabyte verfügt. In Sachen Kompatibilität ist der TurbosPORT-386 mit dem Z-248 vergleichbar, abgesehen vom 8/16-Bit AT-Bus. Die Grundausstattung des TurbosPORT-386 umfaßt ein 3,5"-Diskettenlaufwerk mit einer Kapazität von 1,44 Megabyte. Außerdem ist die größere Festplatte mit 40 Megabyte im Lieferumfang enthalten. Bei der Videoschnittstelle handelt es sich um ein CGA-kompatibles LCD-Display. Zusätzlich ist ein externer Videoport vorhanden, an den herkömmliche Monitore angeschlossen werden können. Das LCD-Display ist schwenkbar in einem Bereich von 180 Grad, so daß ein externer Monitor direkt auf dem Rahmen des Displays abgestellt werden kann. Zusammen mit der Möglichkeit, eine größere Tastatur anzuschließen, ist der TurbosPORT-386 damit grundsätzlich auch als Desktop-Gerät verwendbar.

Das LCD-Display besitzt gegenüber Plasma-Bildschirmen den Vorteil, daß es netzunabhängig, also mit Batterien, betrieben werden kann. Zum TurbosPORT-386 und zum SupersPORT-286 sind deshalb auch wiederaufladbare NiCd-Batteriesätze erhältlich, die einen netzunabhängigen Betrieb ermöglichen. Dabei ist die Akku-Box beim TurbosPORT-386 fest in das Gerät integriert, während sie beim SupersPORT-286 abnehmbar ist. Während des Netzbetriebs werden die Akkus, die gegen Überladung geschützt sind, automatisch wieder aufgeladen.

Der TurbosPORT-386 besitzt einen INTEL 80386-Prozessor, der mit 12 bzw. 6 MHz bei 0 Wartezyklen betrieben wird. Die Taktfrequenz läßt sich über die Tastatur umschalten. Das ist vor allem für den netzunabhängigen Betrieb von Bedeutung, da bei 6 MHz weniger Energie verbraucht wird. Der interne Datenbus des TurbosPORT-386 ist 16 Bit breit. Man kann hier also von einem 16/32-Bit-Rechner sprechen. Für den mathematischen Co-Prozessor 80387 ist bereits ein Sockel vorhanden.

Der Arbeitsspeicher von 2 Megabyte kann auf 3 Megabyte ausgebaut werden. Bis zu zwei Megabyte werden als Expanded Memory unterstützt. Das bedeutet, daß auch unter MS-DOS die vollen zwei Megabyte Arbeitsspeicher benutzt werden können. Weiterhin verfügt der TurbosPORT-386 über 64 Kilobyte ROM und ein auf 128 Kilobyte erweiterbares Slushware-RAM. Das Gerät besitzt wie üblich sowohl eine serielle als auch eine parallele Schnittstelle. Darüber hinaus verfügt der TurbosPORT-386

über eine batteriegepufferte Echtzeituhr. Zusätzlich läßt sich an dem TurbosPORT-386 eine externe Erweiterungsbox anschließen, die zwei PC-kompatible 8-Bit-Steckkarten aufnehmen kann. Als Betriebssystem unterstützt der TurbosPORT-386 MS-DOS 3.21, Xenix V.5 und MS-OS/2 ab Version 1.0, das von Zenith ausgeliefert wird. Wie man leicht aus der gesamten Architektur des Rechners entnehmen kann, begünstigt er durchweg den Einsatz von MS-OS/2.

Der SupersPORT-286 ist, wie schon erwähnt, der kleinere Bruder des TurbosPORT-386 und ebenso wie dieser voll OS/2-fähig. Er wird in zwei Modellvarianten ausgeliefert. Die eine Variante umfaßt ein 3,5"-Laufwerk mit 1,44 Megabyte sowie eine 20 Megabyte-Festplatte, während in der zweiten Version statt der 20 Megabyte-Platte eine 40 Megabyte-Platte ausgeliefert wird. Tastatur und Videoschnittstelle sind identisch mit dem eben beschriebenen TurbosPORT-386. Wie beim TurbosPORT-386 kann auch zusätzlich ein numerisches Keypad angeschlossen werden, was auch ihn zu einem vollwertigen Bürogerät macht. Beim SupersPORT-286 wird die CMOS-Variante des Prozessors 80286, der 80C286, verwen-

det. Der Energieverbrauch von CMOS-Bausteinen ist wesentlich geringer als bei herkömmlichen Bausteinen, was besonders im netzunabhängigen Betrieb eine Rolle spielt. Deshalb werden z.B. auch Digitalarmbanduhren ausschließlich in CMOS-Schaltungstechnik gefertigt. Der 80C286 wird hier mit 12 oder 6 MHz und einem Wartezyklus betrieben. Für den entsprechenden Co-Prozessor, den 80C287, ist ein Sockel vorhanden. Der Arbeitsspeicher umfaßt in der Grundausstattung 1 Megabyte RAM mit Expanded Memory Support (EMS). Der Arbeitsspeicher ist bis auf 5 Megabyte ausbaufähig. Genau wie beim TurbosPORT-386 verfügt der SupersPORT-286 über 64 Kilobyte ROM und ein bis auf 128 Kilobyte ausbaufähiges Slushware-RAM. Die restlichen Daten stimmen alle mit dem TurbosPORT-386 überein.

Betrachtet man die gesamte Palette der hier beschriebenen Rechner, dann wird deutlich, daß Zenith Data Systems den innovativen Bereich im Einsatz von PC-Hochleistungsrechnern fördert und unterstützt. Zenith hat sehr früh erkannt, daß dazu auch das Betriebssystem MS-OS/2 gehört.

7 Wie das Betriebssystem MS-OS/2 auf dem Rechner eingerichtet ist

Größere Festplatten werden oft in Partitionen unterteilt. Partitionen sind Speicherbereiche auf der Festplatte, die bestimmte Eigenschaften haben. Eine Partition muß es mindestens geben. Sie enthält das Betriebssystem, und von ihr aus wird gebootet. Sie kann derzeit maximal 32 Megabyte groß sein. Das liegt an der Beschränkung des Adreßraumes unter MS-DOS und der 8088/8086-Prozessoren. In der für den Herbst 1988 angekündigten MS-OS/2 Version 1.1 ist die Erweiterung der Größe von Partitionen auf maximal 512 Megabyte vorgesehen.

An der bis jetzt noch vorhandenen Beschränkung der Speichergöße von Partitionen liegt es auch, daß Sie mindestens 2 Partitionen eingerichtet haben müssen, wenn Ihre Festplatte physikalisch mehr als 32 Megabyte Speicher zuläßt und wenn Sie diesen Speicher nutzen wollen. Ein weiterer Grund für mehr als eine Partition ist dann gegeben, wenn Sie mehr als ein Betriebssystem verwenden wollen. Dann müssen Sie in der Regel für jedes Betriebssystem jeweils eine Parti-

tion einrichten. Dank einer Eigenentwicklung der Zenith-Fachleute - dem sogenannten DUAL BOOT SYSTEM - es jedoch möglich, MS-OS/2 und MS-DOS in einer gemeinsam genutzten Partition zu betreiben.

Zenith hat als erster Hardware-Hersteller MS-OS/2 an seine Rechner angepaßt und gleichzeitig das ZDS DUAL BOOT SYSTEM auf den Markt gebracht. Es beruht darauf, daß beide Betriebssysteme, das gewohnte MS-DOS und das zukunftsweisende MS-OS/2, im Rechner installiert sind. Das ist insbesondere für die Übergangszeit - bis alle Software-Hersteller ihre Anwendungssoftware auf den Real-Mode angepaßt haben - eine wichtige Zusatzeinrichtung. Denn sie ermöglicht den nahtlosen Übergang von der zweiten zur dritten Generation der Betriebssysteme für Mikros bzw. für PC's. Dies wird durch die Tatsache bestätigt, daß Microsoft dieses Konzept offensichtlich übernommen hat. Es wurde im ersten Update des SDKs (Software Deve-

lopment Kit) 3-4 Monate später - also im März/April '88 - mit ausgeliefert.

Nach dem Einschalten des Rechners fragt das System ab, ob man mit MS-DOS oder mit MS-OS/2 starten will. (Bild 7.1)

Hat man sich entschieden, indem man die Esc-Taste oder die Return-Taste betätigt, dann sucht das Betriebssystem während des sogenannten Boot-Vorganges nach zwei (bzw. drei) speziellen Dateien:

CONFIG.SYS und AUTOEXEC.BAT für den MS-DOS-Betrieb
bzw.
CONFIG.OS2 und
AUTOEXEC.OS2/STARTUP.CMD für den OS/2-Betrieb

Die Config-Datei wird zuerst gelesen, ihr Text legt die grundsätzliche Konfiguration der benötigten Geräte (Devices) fest. Danach folgt die Autoexec-Datei, die sogenannte Startdatei, die über gewisse Betriebssystembefehle die weitere System-Umgebung für den Betrieb festlegt. Die Beschreibung im einzelnen folgt später in diesem Abschnitt.

MS-OS/2 braucht für beide Betriebsarten jeweils eine eigene Startdatei. Die Autoexec.os2 entspricht der Autoexec.bat von MS-DOS 3.x und dient dem kompatiblen Real-Mode. Die neue Startdatei STARTUP.CMD wird von CMD.EXE, dem neuen Befehlsinterpreter, im Protected-Mode der Multitasking-Betriebsart aufgerufen.

Wenn das Betriebssystem diese Dateien oder auch nur eine von ihnen nicht findet, dann startet der Computer mit einer Minimalkonfiguration. Unter Umständen kann man dann bestimmte Teile der Hardware nicht verwenden bzw. das System verhält sich anders als erwartet.

Wir wollen hier nur je eine Variante der genannten Dateien beschreiben, wie sie heute gang und gäbe sind und wie wir sie für optimal halten. Im Ausgang davon können Sie die Dateien selbst ergänzen, so daß die Hardware, die in Ihrem Rechner vorhanden ist, voll unterstützt wird.

Zur Erstellung der Dateien können Sie jeden Texteditor verwenden, der Texte im ASCII-Format (ohne spezielle Druckformate) bearbeiten kann. Grundsätzlich steht Ihnen vom Betriebssystem der Editor EDLIN zur Verfügung.

Wir fassen mich bei der folgenden Beschreibung der Befehle kurz, denn fast alles ist aus MS-DOS bekannt. Wenn Sie erst jetzt in die Arbeit mit Personal-Computern einsteigen und alles neu für Sie ist, dann lesen Sie das folgende unter dem Gesichtspunkt, daß Sie sich nicht darum kümmern müssen, weil Ihr Rechner vom Händler korrekt eingestellt (konfiguriert) ist.

WICHTIG zur Schreibweise: In den eckigen Klammern stehen allgemeine Kennzeichnungen für Eingaben, die Sie machen können, aber nicht müssen. Zum Beispiel bedeutet [Laufwerk:], daß Sie an dieser Stelle ein Laufwerk - in der Form A:, B:, C: oder D: - angeben können, wenn Sie dies wollen oder brauchen.

Beispiel für config.sys:

```
FILES = 20
BUFFERS = 22
DEVICE = C:\word\mouse.sys
```

Die Bedeutung der auftretenden Befehle ist wie folgt:

FILES = n

Mit der Zahl n - in unserem Falle mit 20 - legen Sie die maximale Anzahl gleichzeitig geöffneter Dateien

```
ZDS Dual-Boot System
(C) Copyright Zenith Data Systems Corporation 1987.

Select System To Boot:
  Press ENTER to boot MS OS/2
  Press ESC to boot MS-DOS
```

Bild 7.1

(files) fest. Die Voreinstellung ist 8, denn MS-DOS benötigt schon für sich intern 5 geöffnete Dateien. Der Benutzer hat dann noch drei zu seiner Verwendung. Man beachte, daß bestimmte Software - das ist dann angegeben - eine bestimmte Anzahl von geöffneten Files benötigt. Besonders wichtig ist dies bei Datenbank-Sprachen und Netzwerk-Software.

BUFFERS = n

Mit Angabe der Zahl *n* legen Sie die Anzahl von Puffern fest, die MS-DOS während der Diskettenoperationen zur Datenspeicherung zur Verfügung stehen. Puffer sind Zwischenspeicher für Daten, die von der Festplatte oder den Disketten gelesen werden. Der Standardwert für *n* ist 3. Eine gültige Eingabe ist eine Zahl zwischen 1 und 100.

In unserem Beispiel sind 22 Puffer installiert, was für MS-DOS eine sinnvolle und in den meisten Fällen gut ausreichende Anzahl ist.

DEVICE = [Laufwerk:] [Pfad] Dateiname [.erw]

Device ist ein Befehl, der Ihrem System die Geräte zuweist, die Sie benutzen (betreiben) wollen. Dazu müssen gewisse Programme aufgerufen werden, die man auch Treiberprogramme nennt. In unserem Fall wird das Treiberprogramm für die Maus-Steuerung geladen. Mit DEVICE können Sie also spezielle Hardware, zum Beispiel die MICROSOFT-Maus, installieren.

Sollten Sie keine Microsoft-Maus zur Verfügung haben, lassen Sie den letzten Befehl (DEVICE = ...) in der CONFIG-Datei weg.

Mit den Zusätzen [Laufwerk:] [Pfad] geben Sie das Verzeichnis Ihrer Festplatte oder Diskette (die Adresse) an, wo sich Ihr Treiberprogramm befindet - zum Beispiel in c:\mouse. Wenn Sie also den Befehl verwenden, müssen Sie vor dem nächsten Systemstart die Datei "MOUSE.SYS" von Ihrer Microsoft-Diskette in dieses Verzeichnis kopieren. Im einfachsten Falle - das ist auch unser Beispiel - kopieren Sie in die Wurzel der Festplatte oder auf die Bootdiskette, dann braucht nichts weiter angegeben zu werden.

Beispiel für autoexec.bat:

```
PATH = C:\SYSTEM;c:\rbsys;c:\word;c:\mp;c:\chart;  
c:\win;c:\excel
```

```
ZKEYBGR  
PROMPT $p$g  
DATE  
TIME
```

Der Befehl PATH definiert einen Suchpfad für Programme. Die allgemeine Schreibweise für diesen Befehl ist folgende:

path [[d:] [Pfad] [;] ...]

Was in eckiger Klammer steht kann dem Befehl hinzugefügt werden, muß aber nicht. d: steht für das Laufwerk - in unserem Fall für die Festplatte c:, Pfade bezeichnen Abteilungen auf der Festplatte und werden immer in der Form z.B. \system geschrieben. Hier ist \system die Abteilung, wo MS-DOS untergebracht ist. Mehrere Verzeichnisse werden durch ein Semikolon getrennt. Klein- oder Groß-Schreibung ist übrigens für MS-DOS und für MS-OS/2 gleichgültig.

Das path-Kommando teilt dem System mit, in welchen Verzeichnissen nach Programmen gesucht werden soll, wenn diese nicht im aktuellen Verzeichnis gefunden werden. Ein Suchpfad ist solange gültig, bis ein neuer Pfad angegeben wird.

MS-DOS ebenso wie OS/2 sucht die Pfade in der Reihenfolge ab, in der sie im path-Kommando angegeben werden. Wird ein Programm aufgerufen, das in mehreren Verzeichnissen existiert, so wird zur Ausführung die erste Kopie verwendet, die im Pfad gefunden wird.

Wenn das path-Kommando ohne Optionen eingegeben wird, wird der aktuelle Pfad angezeigt.

Der Befehl ZKEYBGR leistet hier das gleiche wie ein KEYBGR in MS-DOS, nämlich die Umschaltung auf die deutsche Tastatur. Das "Z" in ZKEYBGR macht lediglich deutlich, daß die Zenith-Version von MS-DOS verwendet wird.

Das PROMPT-Kommando bewirkt eine bestimmte Form des Symbols, das signalisiert, daß Sie einen DOS-Befehl eingeben können. Für MS-DOS hat das System ebenso wie später für den Real-Mode als Standardwert den Laufwerksbuchstaben gefolgt von einem ">"-Zeichen (z.B. "C>"). Für den Protected-Mode wird als Prompt-Symbol der aktuelle Laufwerksname, der in eckige Klammern eingeschlossen wird, verwendet (z.B. "[C:\]").

Verschiedene "\$x"-Zeichenfolgen können im Prompt-Kommando benutzt werden, um eigene Aufforderungssymbole zu erzeugen. In unserem Fall wurden nur zwei verwendet:

\$p bewirkt, daß das aktuelle Verzeichnis auf dem Standardlaufwerk angezeigt wird. Z.B. so:
C:\system>. Und \$g produziert das ">"-Zeichen.

Der DATE-Befehl erlaubt die Eingabe eines Datums oder die Veränderung des dem System bekannten Datums. Beim Ablauf der Autoexec.bat wird der Benutzer aufgefordert, das Datum einzugeben oder zu korrigieren. Soll das Datum nicht geändert werden, genügt es, die Eingabetaste zu betätigen.

Der TIME-Befehl funktioniert analog zum DATE-Befehl und erlaubt die Eingabe oder die Änderung der Systemuhrzeit. Die Zeitinformation wird bei der Aktualisierung eines Verzeichnisses verwendet, wenn Dateien erstellt oder verändert werden. Die Eingabe von TIME ohne Optionen zeigt die aktuelle Zeit an und ermöglicht eine Änderung. Soll die Zeit nicht geändert werden, ist lediglich die Eingabetaste zu betätigen. Die Angabe eines neuen Werts muß im 24-Stunden-Format erfolgen - z.B. so: 13:37:56.

Soweit zu den MS-DOS-Dateien, die für Konfiguration und Start beim DUAL-BOOT-SYSTEM notwendig sind. Kommen wir nun zu MS-OS/2. Sie benötigen für beide Betriebsarten, Real und Protected-Mode, nur eine CONFIG.OS2 Datei. Einige Befehle sind jedoch nur im Real-Mode wirksam. Wir werden Sie entsprechend kennzeichnen.

Beispiel für config.os2:

```
FILES = 20
BUFFERS = 50
PROTSHELL = shell.exe cmd.exe /k \initenv.cmd
SHELL = \os2\rbin\command.com c:\os2\rbin /p
RMSIZE = 640
PROTECTONLY = NO
DEVICE = c:\os2\dev\ega.sys
DEVICE = c:\os2\dev\ansi.sys
DEVINFO = scr,ega,c:\os2\dev\viotbl.dcp
DEVINFO = kbd,gr, c:\os2\dev\keyboard.dcp
CODEPAGE = 437,850
LIBPATH = C:\os2\lib
DEVICE = \os2\dev\pointdd.sys
DEVICE = \os2\dev\mouse02.sys,serial=com1
RUN = \os2\pbin\spool.exe C:\os2\spool /o:lpt1
```

Die meisten Befehle sind schon von MS-DOS bekannt. Sie bedeuten im einzelnen:

PROTSHELL = [d:] [Pfad] Dateiname [.erw] [Argument]

Mit dem PROTSHELL-Befehl legen Sie den Befehlsinterpreter des Protected-Mode fest (Standardeinstellung: cmd.exe). "Dateiname" bezeichnet den Dateinamen der Session-Manager-Shell. Dagegen bezeichnet Argument das Laufwerk, den Pfad und den Dateinamen des Befehlsinterpreters für den Protected-Mode.

SHELL = [d:] [Pfad] Dateiname

Das SHELL-Kommando lädt und startet einen Befehlsinterpreter (ein Programm, das im Hintergrund abläuft, die Tastatur bedient und Eingaben von der Tastatur verarbeitet) im angegebenen Laufwerk und Unterverzeichnis.

Dieser Befehl ist zum Beispiel für Systemprogrammierer sinnvoll, die Ihren eigenen Befehlsinterpreter geschrieben haben.

Der SHELL-Befehl bezieht sich nur auf den Real-Mode. Der entsprechende Befehl für den Protected-Mode ist PROTSHELL.

Beispiel: Start des Befehlsinterpreters
COMMAND.COM im Unterverzeichnis OS2

SHELL = \OS2\COMMAND.COM
RMSIZE = n

Mit dem RMSIZE-Befehl legen Sie die Größe des Speicherbereichs fest, der für Real-Mode-Anwendungen zur Verfügung gestellt wird. Die automatische Einstellung des Parameters n, die MS-OS/2 beim Start Ihres Computers festlegt, hängt von der Größe des installierten RAM-Speichers ab.

Wenn Ihr Computer über 640 Kilobyte RAM verfügt, dann berechnet sich die Zahl n aus 640 minus der für Protected-Mode mindestens erforderlichen Speichergröße. Sind insgesamt 1 Megabyte oder mehr an RAM installiert, wird für Real-Mode-Anwendungen der Speicher reserviert, der sich unterhalb der 1-Megabyte-Grenze befindet, also 512 oder 640 Kilobyte.

Angegeben wird die Speichergrenze in ganzen Zahlen von 0 bis 640 Kilobytes.

Beispiel: Sie wollen 512 Kilobyte für Real-Mode-Anwendungen reservieren. Dazu tragen Sie die folgende Zeile in Ihre CONFIG.SYS-Datei ein:

```
RMSIZE = 512  
PROTECTONLY = [YES/NO]
```

Diese Parameter-Einstellung legt den Modus fest, in dem sich Ihr Computer nach dem Start durch MS-OS/2 befindet.

Mit dem Parameter YES erlauben Sie nur den Ablauf von Programmen im geschützten Mode (Multitasking), mit dem Parameter NO können Sie Programme in beiden Modi ablaufen lassen.

Unser Beispiel soll den Real- und Protected-Mode zulassen. Die Umschaltung von Betriebsart zu Betriebsart erfolgt dann über die Tastatur und durch den Session-Manager, wie weiter unten beschrieben wird. Unsere Einstellung ist also
PROTECTONLY = NO.

```
DEVINFO = Gerätename,Typ,[Pfad]Datei-  
name,[ROM = [(|xxx[,yyy])]]
```

Der DEVINFO-Befehl liefert dem Betriebssystem Informationen über spezielle Geräte, die Sie zusammen mit Ihrem Rechner benutzen. Der DEVINFO-Befehl arbeitet eng mit dem CODEPAGE-Befehl zusammen und erlaubt den Wechsel der Zeichensatz-Tabelle für ein bestimmtes Gerät.

Die Bedeutung der verwendeten Variablen erklärt sich wie folgt:

"Gerätename" beschreibt einen Gerätenamen, für die stellvertretend die Outputstellen stehen: PRN, LPT1, LPT2, LPT3, KBD\$ oder SCREEN\$.

"Dateiname" benennt die Datei, die die Zeichensatz-Tabellen für den Bildschirm und den Drucker enthält. Dateiname kann aber auch die Umsetzungstabellen (engl. translation tables) für die Tastatur bezeichnen.

"ROM" ist ein optionales Schlüsselwort, mit dem Sie ein oder zwei Zeichensatz-Tabellen (Code pages) angeben können, die im ROM des Geräts enthalten bzw. dem Gerät über eine Cartridge verfügbar sind. Diese Option kann jedoch nur für SCR, PRN, LPT1, LPT2 und LPT3 verwendet werden.

xxx,yyy bezeichnen jeweils Nummern für eine Zeichensatz-Tabelle.

Bei Druckern kann jede ROM-Angabe aus einem Wertepaar, das in Klammern eingeschlossen wird, bestehen. Dabei bezeichnet der erste Wert die Zeichensatz-Tabelle und der zweite Wert die Font-Nummer für das ROM bzw. die Cartridge, die der Drucker benutzt. Standardmäßig wird die Font-Nummer 0 angenommen.

Beispiel: Wenn Sie für Ihren EGA-Monitor eine Zeichensatz-Umschaltung vorsehen wollen, dann könnte der entsprechende DEVINFO-Befehl ungefähr so aussehen:

```
DEVINFO = SCR , EGA , monitor.tbl  
CODEPAGE = xxx [,yyy]
```

Dieser Befehl ermöglicht Ihnen, die Zeichensatz-Tabellen (Code Page) auszuwählen, zwischen denen Sie dann wechseln können. Eine Zeichensatz-Tabelle beschreibt den Zeichensatz, den Sie benutzen. Unter einem Zeichensatz versteht man hier eine Menge von landes- oder sprachspezifischen Sonderzeichen. Im deutschen Zeichensatz sind das u.a. die Umlaute ä, ö und ü oder auch das ß.

Die Zahlen xxx und yyy repräsentieren zwei Zeichensatz-Tabellen, die das System unterstützt. Von diesen beiden Zeichensatz-Tabellen sollen nur beispielhaft genannt werden:

437 für USA
850 für Mehrsprachig
860 für Portugiesisch
863 für Franz. - Kanadisch
865 für Nordisch

Für jedes Gerät, das mit den ausgewählten Zeichensatz-Tabellen arbeiten soll, müssen Sie einen entsprechenden DEVINFO-Befehl in die CONFIG.SYS-Datei aufnehmen.

Beispiel: Wenn Sie den Befehl "CODEPAGE = 860,850" verwenden, dann unterstützt MS-OS/2 alle Anwendungen, die einen dieser beiden Zeichensätze benutzen.

```
LIBPATH = d: Pfad [ d: Pfad] [...]
```


Mit dem LIBPATH-Befehl geben Sie dem Betriebssystem die Verzeichnisse an, in denen es nach Dynamic-Link-Modulen suchen soll. Dynamic-Link-Module werden zur Laufzeit eines Programms, das heißt dynamisch geladen. Deshalb müssen die Verzeichnisse, in denen sich diese Modulbibliotheken (libraries) befinden, dem System vorher bekannt sein. Ansonsten nimmt MS-OS/2 den Pfad "C:\\" als Standardwert an.

Wie beim PATH-Befehl können Sie mehrere Pfade angeben, die durch ein Semikolon getrennt werden müssen. Im Gegensatz zum PATH-Befehl wird jedoch nicht zunächst das aktuelle Verzeichnis durchsucht, sondern nur die im LIBPATH-Befehl explizit angegebenen Verzeichnisse.

Beispiel: Wenn Sie Ihre Dynamic-Link-Module im Verzeichnis "C:\LIB" und nicht im Wurzelverzeichnis speichern wollen, dann müssen Sie die folgende Zeile in Ihre CONFIG-SYS-Datei einfügen, um Ihren Programmen den Zugriff auf die Modulbibliothek zu ermöglichen:

```
LIBPATH = C:\LIB
RUN = [d:] [Pfad] Programmname [.crw] [Argumente]
```

Mit dem RUN-Befehl können Sie während des Systemstarts ein Anwender-Programm oder einen Systemprozeß starten. Der RUN-Befehl kann auch mehrfach in einer CONFIG.SYS Datei vorkommen.

Beispiel: Sie wollen ein Weckprogramm installieren, daß jeweils zu einer bestimmten Zeit einen Alarm auslöst. Stunde und Minute des Alarms werden dem Programm in der Befehlszeile mitgeteilt. Das könnte dann ungefähr so aussehen:

```
RUN = Wecker 12 55
```

Während des Systemstarts sucht MS-OS/2 außer nach der Datei CONFIG.OS2 noch nach der Datei AUTOEXEC.OS2 für den Real-Mode bzw. nach der Datei STARTUP.CMD für den Protected-Mode.

Die Datei AUTOEXEC.OS2 enthält eine Reihe von Anweisungen und Programmaufrufen, wie sie - im Gegensatz zu CONFIG.OS2 - auch auf der Befehls-ebene unter MS-OS/2 eingegeben werden können.

AUTOEXEC.OS2 wird nur im Real-Mode wirksam. Diese Datei ist also nur dann wirksam, wenn Sie im CONFIG.OS2 den Parameter PROTECTONLY auf NO gesetzt haben.

Wenn die Befehle DATE und TIME im AUTOEXEC.OS2 fehlen, dann wird Datum und Zeit nicht gezeigt, und Sie werden auch nicht nach dem aktuellen Datum und der Uhrzeit gefragt. Tragen Sie stets die korrekten Angaben ein, denn MS-OS/2 benutzt diese Daten, um die Einträge in den Inhaltsverzeichnissen der Festplatten und Disketten zu aktualisieren.

An dieser Stelle soll uns wieder eine einfache Beispieldatei genügen, die in elementarer Weise den meisten Ansprüchen gerecht wird.

Beispiel für autoexec.os2:

```
ECHO OFF
DATE
TIME
PROMPT $p$g
PATH C:\system;C:\os2;C:\OS2\BIN;C:\OS2\RBIN
CLS
VER
```

Die verwendeten Befehle sind allesamt aus den MS-DOS Versionen bekannt. Kurz gesagt macht unser Beispiel eines Programmes aus MS-OS/2-Befehlen im Real-Mode folgendes: Die Echo-Meldungen des Betriebssystems werden ausgeschaltet, Datum und Zeit gezeigt, die Aufforderung zur Befehlseingabe (PROMPT) auf Laufwerk, Unterverzeichnis (Directory) und ">" gesetzt - zum Beispiel C:\WORD>. Dann wird der Pfad (PATH) festgelegt, auf dem nach Programmen gesucht wird, die sich nicht im aktuellen Verzeichnis befinden. Schließlich wird wieder der Bildschirm freigemacht und die Version des Betriebssystems angezeigt (VER).

Im Aufbau gibt es keinen Unterschied zwischen den beiden Startdateien autoexec.os2 und startup.cmd. Sie greifen nur auf die verschiedenen Befehlsvorräte zu, wie sie durch den COMMAND.COM und den CMD.EXE gegeben sind. Natürlich lösen sie dementsprechend - und wie schon erwähnt - verschiedene Aufgaben: die eine beschreibt den Startlauf für den Real-Mode, die andere für den Protected-Mode. Eine einfache STARTUP.CMD könnte wie folgt aussehen:

Installation

Beispiel für startup.cmd:

```
ECHO OFF
PATH C:\;C:\OS2\BIN;C:\OS2\PBIN
```

Wenn Sie die deutsche Tastatur nicht über die config.sys-Datei einstellen wollen, dann müssen Sie außerdem die folgenden Befehle hinzufügen:

```
DPATH c:\os2\dev
KEYB GR
```

Mit dem path-Befehl legen Sie einen Suchpfad für Programmdateien und mit dem dpath-Befehl einen entsprechenden Suchpfad für Datendateien. Immer wenn MS-OS/2 irgendeine Datei im aktuellen Verzeichnis nicht vorfindet, dann sucht es auf dem vorgezeichneten Weg erst in der Wurzel (C:\), dann im Verzeichnis C:\OS2\BIN und schließlich in C:\OS2\PBIN. dpath ist nur im Protected-Mode verwendbar, im Real-Mode müßte hier der APPEND-Befehl stehen.

Schließlich kommt der neue Aufruf der deutschen Tastatur keyb gr, für dessen erfolgreichen Ablauf hier die Pfade gelegt worden sind. KEYB ist in MS-OS/2 ein Befehl, und GR bezeichnet den deutschen Zeichensatz, der - wie die übrigen Zeichensätze auch -

in der Datei KEYBOARD.DCP abgelegt ist. Diese Datei befindet sich im Verzeichnis c:\os2\dev. Ver zeigt die laufende Version an.

Beide Dateien, CONFIG.OS2 und STARTUP.CMD bzw. AUTOEXEC.OS2, werden während des Systemstarts automatisch geladen und die hierin gespeicherten Befehle ausgeführt - zuerst werden die CONFIG-Parameter gesetzt und dann die AUTOEXEC-Befehle ausgeführt. Während im AUTOEXEC.BAT bzw. im STARTUP.CMD allgemeine MS-OS/2-Befehle verwendet werden und die Ausführung daher auch nur jeweils im Real oder Protected-Mode erfolgt, können im CONFIG.OS2 nur ganz spezielle Anweisungen erteilt werden.

Ein weiterer Unterschied besteht darin, daß Sie Ihren Computer neu booten müssen, um Änderungen im CONFIG.OS2 wirksam werden zu lassen. Dagegen kann die Datei AUTOEXEC.OS2 aus dem Real-Mode in der Befehlsebene durch einfachen Aufruf gestartet werden:

AUTOEXEC <Return>

oder entsprechend im Protected-Mode

STARTUP <Return>

```
C:\>PATH=C:\SYSTEMS;c:\rbsys;c:\word;c:\chart;c:\win;c:\excel
C>ZKEYBGR
                                ZKEYBGR Version 3.21D
                                Copyright (C) 1987, Zenith Data Systems GmbH

C>prompt $p$g
C:\>verify on
C:\>DATE
Current date is Sun 5-29-1988
Enter new date (mm-dd-yy):

C:\>TIME
Current time is 12:44:53.03
Enter new time:

C:\>
```

Bild 7.2

Sie sollten berücksichtigen, daß sich Ihr Rechner während des Systemstarts "aufhängen" kann, wenn in der Datei CONFIG.OS2 oder AUTOEXEC.OS2 fehlerhafte Eingaben gemacht wurden. In diesem Fall gehen Sie am besten folgendermaßen vor:

1. Schalten Sie den Rechner aus
2. Legen Sie die Kopie Ihrer "MS-OS/2 Program Disk" in das Laufwerk A: ein

3. Schalten Sie Ihren Rechner ein und verriegeln Sie das Laufwerk A:

Nach der Systemmeldung können Sie auf die Festplatte C: wechseln und die Dateien erneut bearbeiten. Starten Sie danach Ihren Computer neu. Zum Beispiel im Real-Mode müßte folgendes Bild erscheinen: (Bild 7.2)

8 Die nachträgliche Installation von MS-OS/2 auf dem Rechner

Wir wollen den Fall beschreiben, daß Sie Ihren Computer ganz neu einrichten wollen. Vor allem soll auch eine neue MS-DOS-Version, zum Beispiel 3.2, installiert werden. Abgesehen vom Betriebssystem sollten Sie alle Ihre Daten und Programme auf Diskette gesichert haben, bevor Sie die folgenden Schritte tun. Wir beschreiben hier lediglich den Weg, der für MS-OS/2 standardmäßig gilt. Zenith bietet neben FDISK das Hilfsprogramm PART an. Beide Programme leisten jedoch sinngemäß das gleiche.

WICHTIG zur Schreibweise: Im folgenden werden Tasteneingaben beschrieben, zum Beispiel die Betätigung der Return-Taste. In diesem Falle schreiben wir <Return>. Das bedeutet, daß Sie die Return-Taste anschlagen müssen. Ein weiteres Beispiel: die Y-Taste für die jeweilige Yes-Eingabe (Ja) wird <Y> geschrieben.

Ziel: Sie wollen eine DOS-Partition für die neue Version von MS-DOS anlegen.

```

Personal Computer
Fixed Disk Setup Program Version 1.00

FDISK Options

Choose one of the following:

    1. Create Microsoft Operating System/2 partition
       or logical drive
    2. Change the active partition
    3. Delete Microsoft Operating System/2 partition
       or logical drive
    4. Display partition data

Enter choice: [1]

Press Enter to continue or
Esc to return to Microsoft Operating System/2

```

Bild 8.1

Weg: Booten Sie Ihren Computer mit der MS-DOS-Programmdiskette (Program Disk Nr.1) und rufen Sie dann das Hilfsprogramm FDISK auf, wenn kein gesondertes Install-Programm vorliegt. Schließen Sie jeden eingetippten Befehl ab, indem Sie die Return-Taste (Enter-Taste) betätigen.

FDISK <Return>

FDISK legt den Speicherbereich für das Betriebssystem physikalisch, das heißt auf der Hardware-Seite fest. Das Hilfsprogramm meldet sich zu diesem Zweck mit einem Menü, das aus vier Punkten besteht. Wenn Sie mehr als eine Festplatte in ihrem System installiert haben, erscheint ein fünfter Menüpunkt: Nächste Festplatte auswählen (Bild 8.1).

Wählen Sie Menüpunkt 1: Create DOS Partition
(DOS-Partition machen)

Wenn jetzt der Text "Fixed disk already has a DOS partition" (Die Festplatte hat bereits eine DOS-Partition) erscheint, dann betätigen Sie zweimal die <ESC>-Taste, um das Programm FDISK zu verlassen. Dann geht es mit dem Formatieren weiter.

Do you wish to use the entire fixed disk for DOS?
(Y/N)....(Y)
(Wollen Sie die ganze Festplatte unter DOS benutzen?).

Beantworten Sie die Frage mit <Y> oder <Return>. Nach einer kurzen Zeit erscheint die Systemmeldung

System will now restart.....
(Das System startet jetzt neu)

Das System muß noch einmal gestartet werden. Drücken Sie eine beliebige Taste und warten Sie, bis das Betriebssystem erneut von der Diskette geladen wurde. Danach muß die Festplatte formatiert werden.

Bevor Sie formatieren, denken Sie bitte daran, daß alle Daten und Programme, die sich noch auf der Festplatte befinden, unwiderruflich gelöscht werden.

Dieser Vorgang ist nur einmal, beim Einrichten eines neuen Betriebssystems notwendig und sollte im allgemeinen später nicht mehr ausgeführt werden.

Ziel: Nun muß die partitionierte Festplatte formatiert werden, das ist ein Vorgang, bei dem - bildhaft ausgedrückt - die leeren Seiten eines noch vollzuschreibenden Buches festgelegt werden.

Weg: Nach der Systemmeldung A> geben Sie folgenden Befehl ein:

FORMAT C:/V/S <Return>

Das Formatierprogramm fragt noch einmal, ob Sie dies auch wirklich wollen. Beantworten Sie die Frage mit <Y> für Ja.

Das Formatieren dauert etwa zwei Minuten, je nach Größe und Art der Festplatte. Nach erfolgreichem Ablauf können Sie der Festplatte einen Namen geben. Hierzu haben wir den aus früheren DOS-Versionen bekannten Parameter /V eingegeben. Es erscheint am Bildschirm die Meldung:

Volume label (11 characters, ENTER for none)?
(Laufwerksname; 11 Buchstaben; Return-Taste, wenn kein Name gewünscht ist)

Wenn Sie keinen Namen vergeben wollen, geben Sie <Return> ein, andernfalls den Namen, den Sie vergeben wollen. Er darf maximal 11 Buchstaben lang sein - zum Beispiel OS2PLATTE.

Mit dem Parameter /S wurden nun die verdeckten System-Teile des Betriebssystems auf der Festplatte an ganz bestimmter Stelle untergebracht. Es handelt sich dabei um das BIOS und das MSDOS, das sind die grundlegenden INPUT-OUTPUT- und DISK-OPERATION-Systeme. FORMAT adressiert die Sektoren, bildet ihre Verkettung und bestimmt damit von der Software-Seite her die Speicherquantität der Festplatte. Außerdem wird die Spur für das Inhaltsverzeichnis (DIRECTORY) festgelegt. Die Festplatte ist nun für die Aufnahme von Daten und Programmen vorbereitet.

Im nächsten Schritt müssen die Dateien der internen und externen Befehle von MS-DOS auf die Festplatte gebracht werden. Legen Sie dazu die MS-DOS-program-disk in das Laufwerk A: Ihres Computers und geben Sie den nachfolgenden Befehl ein:

COPY *.* C: <Return>

Nachdem die erste Diskette kopiert ist, legen Sie auch noch die zweite MS-DOS-Diskette ins Laufwerk A: und kopieren Sie diese auf die gleiche Art. Unerlässlich ist eigentlich nur die Datei COMMAND.COM, das ist der Befehlsinterpreter von MS-DOS. Der restliche Inhalt der Disketten besteht aus Hilfsprogrammen - insbesondere aus den externen Befehlen. Von nun an brauchen Sie nur noch mit der Festplatte zu arbeiten.

Nun haben Sie Ihr neues MS-DOS installiert. Das war der erste Schritt, der auch bei der Doppel-Installation von den beiden Betriebssystemen MS-DOS und MS-OS/2 im DUAL-BOOT-System nötig ist.

Ziel: Nun geht es darum, auch noch MS-OS/2 auf den Rechner zu bringen. Das geht zum Glück einfacher und schneller. Denn dafür gibt es ein Install-Programm sowohl in der Original-Microsoft-Version als auch in der Zenith-Anpassung.

Weg: Rufen Sie einfach das Install-Programm von dem Laufwerk A: aus auf.

a:Install

Sie werden noch gefragt, wie Sie das DUAL-BOOT anlegen wollen. Wählen Sie die doppelte Start-Möglichkeit und schon sind Sie fertig.

9 Von MS-DOS in vier Stunden nach MS-OS/2

Wenn Sie schon mit einem PC und mit MS-DOS vertraut sind, dann brauchen Sie wirklich nur einen halben Tag, um sich in die Bedienung von MS-OS/2 einzuarbeiten. Dazu können Sie folgendermaßen vorgehen:

1. Hintergrundwissen: Machen Sie sich klar, was Multitasking auf einem Rechner bedeutet. Das ist nicht unwichtig, weil Sie nur so verstehen können, was Sie machen, wenn Sie den Programm-Selektor bedienen. Der Programm-Selektor ist die zeichen-orientierte Bedieneroberfläche, die derzeit bis zur Fertigstellung des Windows-Presentation-Managers MS-OS/2 ausgeliefert wird.

2. Programm-Selektor-Bedienung: Der Programm-Selektor repräsentiert nach außen den tiefer liegenden Session-Manager.

3. Multitasking Befehle: detach, exit

Zum ersten Punkt wird der laufende Abschnitt das wichtigste zusammenfassen und die neuen Fachausdrücke erklären. Dem zweiten Punkt dient der nächste Abschnitt und schließlich informiert Sie der 12. Abschnitt über den dritten Punkt.

MS-OS/2 ist ein Multitasking-Betriebssystem. Multitasking bedeutet, daß mehrere Anwendungen zeitlich verzahnt auf einem Rechner ablaufen können. Dabei wird der Eindruck erweckt, daß jeder Anwendung der gesamte Rechner zur Verfügung steht. Eine Anwendung kann wiederum aus mehreren eigenständigen

Programmen (tasks) bestehen, die miteinander kommunizieren.

Mit Multitasking eröffnen sich eine Reihe von neuen Möglichkeiten am Personal-Computer. Sie brauchen zum Beispiel eine laufende Anwendung nicht mehr zu unterbrechen, um eine neue zu beginnen. Sie können einen Brief schreiben, während gleichzeitig zeitintensive Berechnungen, die keinen Dialog mit dem Bediener brauchen, im Hintergrund durchgeführt werden. Sie können nun weitaus flexibler mit Ihrem Rechner arbeiten, als Ihnen das bisher möglich war.

Die Grundlage des Multitaskingkonzepts von MS-OS/2 ist ein prioritätengesteuertes Zeitscheiben-Konzept, das durch einen in das Betriebssystem integrierten Scheduler oder Dispatcher realisiert wird. Unter einem Scheduler versteht man den Teil eines Betriebssystems, der den einzelnen Tasks ihre Rechenzeit zuweist und den Wechsel von einer Task zur nächsten, den sogenannten Kontextwechsel, durchführt. Dieser Kontextwechsel ist sowohl für den Benutzer als auch für die Tasks selber nicht wahrnehmbar.

Der Scheduler teilt die Rechenzeit prioritätengesteuert zu, das heißt Tasks, die eine hohe Priorität besitzen, werden bei der Auswahl bevorzugt. Wenn z.B. eine Task zeitkritisch ist, erhält sie eine höhere Priorität.

Hat der Scheduler aus der Liste der Tasks, die ablaufbereit sind, eine Task ausgewählt, dann erhält sie die Kontrolle über den Prozessor, bis das nächste Zeitintervall abgelaufen ist. Dieses Multitaskingkonzept wird auch als Zeitscheiben-Konzept (engl. time-slicing) bezeichnet.

Wir haben bislang von Prozessen, Prozeduren und Programmen gesprochen, die beim Multitasking parallel ablaufen können. Das MS-OS/2-Multitaskingkonzept unterscheidet aber genauer zwischen drei verschiedenen Ablauftypen: Threads, Prozesse und Bildschirmgruppen. Diese drei Typen stehen in einer hierarchischen Beziehung zueinander. So gibt es zu jedem Thread einen übergeordneten Prozeß und jeder Prozeß ist wiederum einer Bildschirmgruppe zugeordnet (siehe Bild 9.1).

Ein Anwenderprogramm ist zum Beispiel ein Prozeß. Prozesse sind die Einheiten, die auf Systemressour-

cen, wie Dateien, Arbeitsspeicher und Einrichtungen zur Prozeßkommunikation, zugreifen können. Ein Prozeß kann mehrere konkurrierende Tasks enthalten, die hier auch als Threads (sinngemäß: Ausführungspfade) bezeichnet werden. Jeder Thread besitzt seine eigene Priorität. Ein oder mehrere Prozesse sind jeweils in einer Bildschirmgruppe (engl. screen group) zusammengefaßt, sofern sie sich einen Bildschirm als Schnittstelle zum Bediener teilen. Weil es mehrere Bildschirmgruppen geben kann, verfügt jede über einen virtuellen Bildschirm und eine virtuelle Tastatur. Unter einem virtuellen Bildschirm müssen Sie sich einen Speicherbereich im Rechner vorstellen, in dem stets das aktuelle Bild liegt, das der Prozeß tatsächlich am realen Bildschirm zeigen würde, wenn dieser frei wäre. Den physikalischen Bildschirm und die physikalische Tastatur kann ja nur eine Bildschirmgruppe erhalten. Beides wird der jeweils aktiven Bildschirmgruppe zugewiesen. Diese Bildschirmgruppe können Sie mit dem Session-Manager auswählen (Bild 9.1).

Ein Anwenderprogramm unter MS-DOS 2.x oder 3.x ist vergleichbar mit der elementarsten Form eines MS-OS/2-Prozesses. Prozesse sind gegeneinander geschützt. Einem Prozeß sind normalerweise nur die MS-OS/2-Systemfunktionen und seine Kindprozesse, das sind die Prozesse, die er selbst erzeugt hat, bekannt. Deshalb können Prozesse nicht auf Arbeitspeicherbereiche oder die Ressourcen anderer Prozesse, einschließlich der Kindprozesse, zugreifen, es sei denn mit der Einwilligung des anderen Prozesses.

Dem MS-OS/2-Scheduler sind die eigentlichen Prozesse völlig unbekannt. Er verteilt vielmehr die verfügbare Rechenzeit auf die ausführbaren Tasks innerhalb eines Prozesses, die hier als Threads bezeichnet werden. Ein Thread wird dem Scheduler gegenüber charakterisiert durch seine Priorität und alle Angaben, die den momentanen Stand seiner Bearbeitung beschreiben.

Jeder Prozeß besitzt zunächst nur einen Thread, dem das Betriebssystem die Kontrolle übergibt. Dieser Thread kann nun weitere Threads innerhalb des Prozesses erzeugen. Diese Threads laufen dann asynchron ab, das heißt sie werden parallel und unabhängig voneinander im Rahmen des Multitasking abgearbeitet. Sie können verschiedene Prioritäten haben und ihre Prioritäten ändern. Über diese Möglichkeiten wird realisiert, was wir oben programm-in-

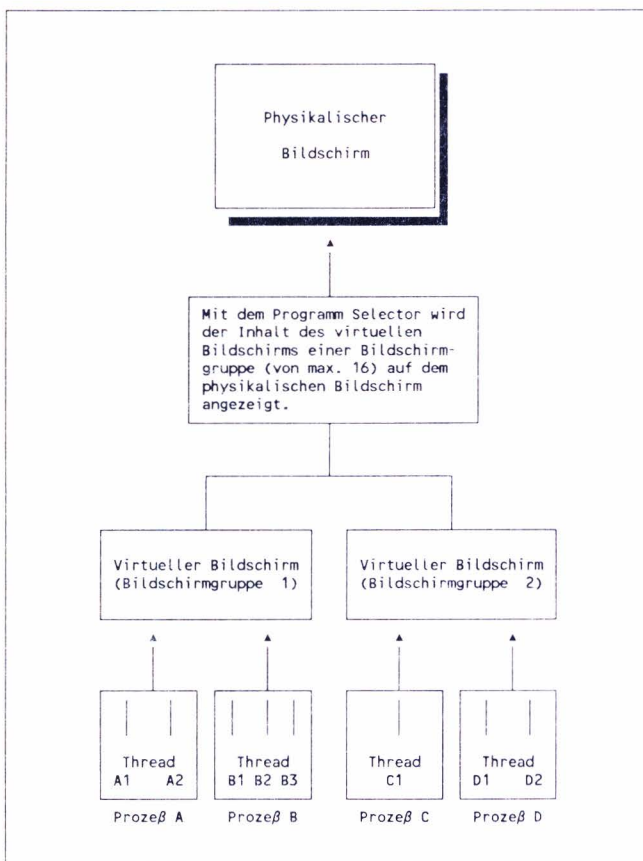


Bild 9.1

ternes oder prozeß-internes Multitasking genannt habe.

Alle Threads eines MS-OS/2-Prozesses verfügen gemeinsam über denselben Datenbereich. Der Zugriff auf gemeinsame Daten und Datenstrukturen muß deshalb synchronisiert (das heißt zeitlich aufeinander abgestimmt) werden. Dies geschieht durch die Verwendung gewisser Konzepte der Interprozeß-Kommunikation.

Während der Ausführung eines Threads innerhalb eines Anwenderprogramms arbeitet der Prozessor auf der Anwenderebene, das heißt der Thread darf nur auf die Ressourcen seiner eigenen Prozeßumgebung zugreifen und kann keine Maschinenbefehle durchführen, die der Systemebene vorbehalten sind.

Ein oder mehrere Prozesse werden in einer Bildschirmgruppe zusammengefaßt. MS-OS/2 unterstützt maximal 16 Bildschirmgruppen. Wenn Sie die Tastenkombination <Strg>-<ESC> betätigen, dann gelangen Sie von der aktuellen Sitzung oder Bildschirmgruppe in den Programm-Selektor. Dort können Sie eine bereits bestehende Bildschirmgruppe auswählen oder eine neue Bildschirmgruppe erzeugen.

Für jede Bildschirmgruppe verwaltet MS-OS/2 einen virtuellen Bildschirm, auf den die Ausgaben aller Prozesse einer Bildschirmgruppe geschickt werden. Sobald Sie eine Bildschirmgruppe mit dem Session-Manager auswählen, wird deren virtueller Bildschirminhalt auf dem physikalischen Bildschirm angezeigt. Innerhalb einer Bildschirmgruppe gibt es jeweils nur einen Vordergrundprozeß, das heißt nur dieser Prozeß führt Ausgaben auf den Bildschirm durch oder liest Eingaben von der Tastatur.

Damit sind in aller Kürze die wichtigste Kennzeichen für das Multitasking mit MS-OS/2 teils zusammengefaßt und teils neu dargestellt. Dabei sind einige Male die Schutz-Mechanismen angesprochen worden, die MS-OS/2 für Datenbereiche und Programme bereitstellt. Für den interessierten Leser wollen wir diese Schutz-Mechanismen hier etwas genauer darstellen. In der Debatte um Netzwerke und um die Arbeit an gemeinsamen Daten geht es stets auch um die beiden großen Themen Schutz und Kommunikation. Beides

hängt eng miteinander zusammen: Kein Schutz ohne Kommunikation und keine Kommunikation ohne ausreichenden Schutz.

Der Intel 80286 besitzt im 80286-Protected-Mode vier Schutzebenen. Diese Schutzebenen sind hardwaremäßig realisiert. Im MS-OS/2-Protected-Mode werden von diesen Schutzebenen normalerweise zwei benutzt - die Systemebene (Ebene 0) und die Anwenderebene (Ebene 3). Auf der Systemebene ist eine totale Kontrolle über den Rechner möglich. Diese Ebene ist daher ausschließlich dem Betriebssystem vorbehalten. Mit jeder höheren Stufe nehmen die Steuerungsmöglichkeiten und damit auch die Rechte ab. Deshalb unterliegen Anwenderprogramme einigen Einschränkungen.

Während der Prozessor auf Systemebene arbeitet, können alle verfügbaren Maschinenbefehle ausgeführt, und es kann jede beliebige Speicherstelle adressiert werden. Bevor das Betriebssystem die Kontrolle über den Prozessor an ein Anwenderprogramm übergibt, schaltet es den Prozessor zunächst auf die Anwenderebene um. Wenn auf der Anwenderebene versucht wird, gewisse reservierte Maschinenbefehle auszuführen, wie zum Beispiel einen Zugriff auf eine Ein/Ausgabe-Schnittstelle, dann wird ein Hardware-Interrupt (eine Unterbrechung) ausgelöst und die Kontrolle sofort wieder an das Betriebssystem übergeben.

Auf ähnliche Art und Weise wird der Arbeitsspeicher vor unberechtigten Zugriffen von Anwenderprogrammen geschützt. Wenn ein Anwenderprogramm auf einen Arbeitsspeicherbereich zugreifen will, der ihm nicht zugeordnet ist, dann wird dies von den Hardware-Schutzmechanismen des 80286 registriert. Es wird sofort ein Hardware-Interrupt ausgelöst, der es dem Betriebssystem erlaubt, den Zugriff des Anwenderprogramms zu unterbinden.

Um Ihre im Multitasking ablaufenden Programme zu verwalten, stellt Ihnen MS-OS/2 den sogenannten Session-Manager zur Verfügung. Er ermöglicht Ihnen, mehrere neue Anwendungen gleichzeitig zu starten und zwischen diesen Anwendungen beliebig hin und her zu schalten.

10 Die Benutzeroberfläche: der Programm-Selektor und Session-Manager

Eine Session nennt man eine Arbeitssitzung am PC in einer bestimmten Betriebsart. Besitzt nun ein Computer zwei (oder mehr) Betriebsarten, so ist ganz klar, daß das Betriebssystem einen Hyper-Regler bereitstellen muß, der die Wahl zwischen den Betriebsarten erlaubt. Diesen nennt man in MS-OS/2 Session-Manager. Er wird mit der Tastenkombination <Strg>-<ESC> aufgerufen.

Der Programm-Selektor ist die derzeitige, aber immer noch vorläufige Benutzeroberfläche für MS-OS/2. Sie wird später vom Windows-Presentation-Manager abgelöst werden. Der Programm-Selektor gestattet es Ihnen, Bildschirmgruppen anzulegen, zwischen laufenden Programmen (Multitasking) auszuwählen und Ihre Bildschirmgruppen zu verwalten. Bildschirmgruppen sind Programme oder Programmpakete, die in einem Bildschirm laufen.

Zum Programm-Selektor gehören die folgenden MS-OS/2-Dateien: shell.exe, shell.lib, shell11f.aif und

shell11f.aif. Sie müssen alle in demselben Verzeichnis (Directory) liegen. Dabei ist es gleichgültig, welches Directory das ist.

Shell ist der amerikanische Begriff für Programm-Umgebung. Zu ihr gehören alle fixen Systemeinstellungen, insbesondere die Input/Output-Stellen wie Drucker, Maus etc., die dem Programm zur Verfügung gestellt und vom Programm benötigt werden. Weil vor allem auch der Bildschirm zur Programm-Umgebung gehört, wird häufig auch nur die (Bildschirm-)Bedienoberfläche Shell genannt. So gehört die Shell zur Konfiguration und wird deshalb auch in der Config.sys festgelegt. In dem folgenden Beispiel wird die SHELL, das heißt die Oberfläche, in den Pfad c:\os2 gelegt und es wird ein Stapelprogramm c:\os2\initenv.cmd, das die Umgebung (environment) initiiert (init env), für jede neue Bildschirmgruppe im Protected-Mode aufgerufen:

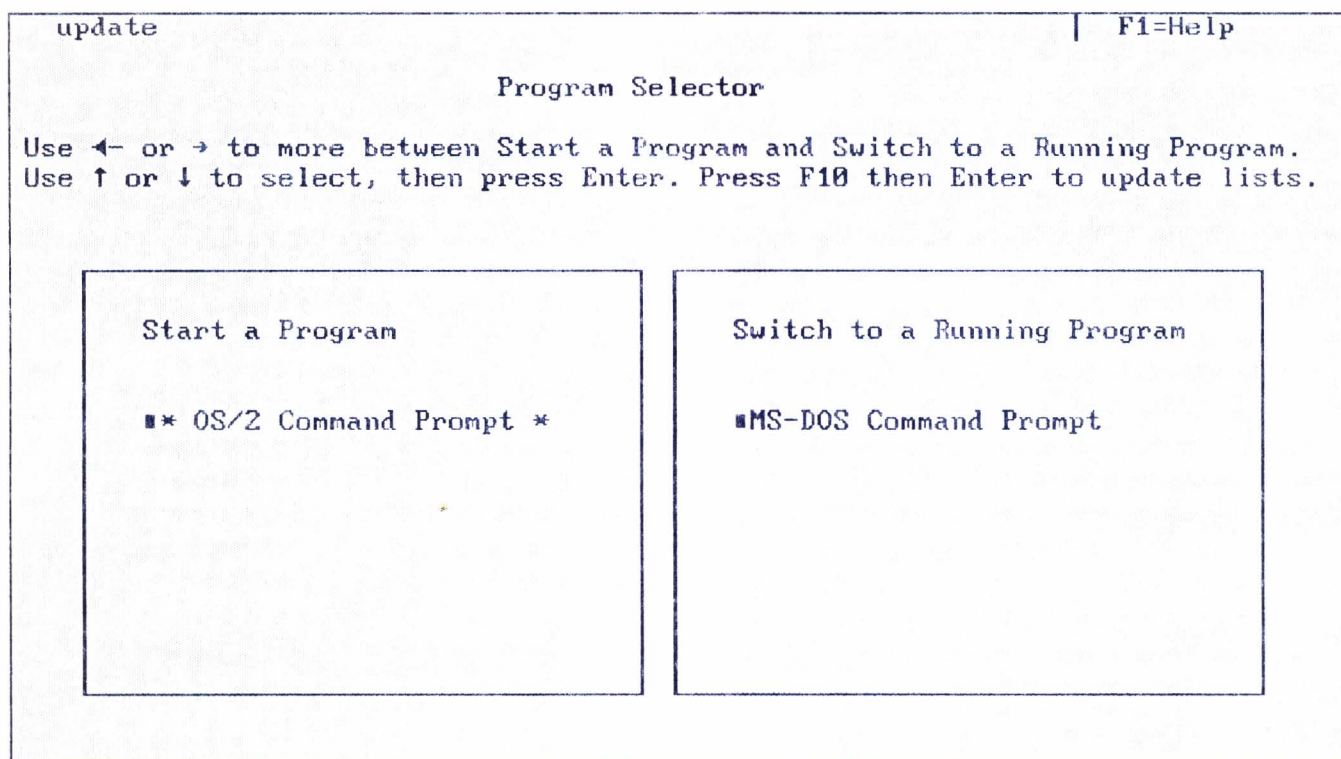


Bild 10.1

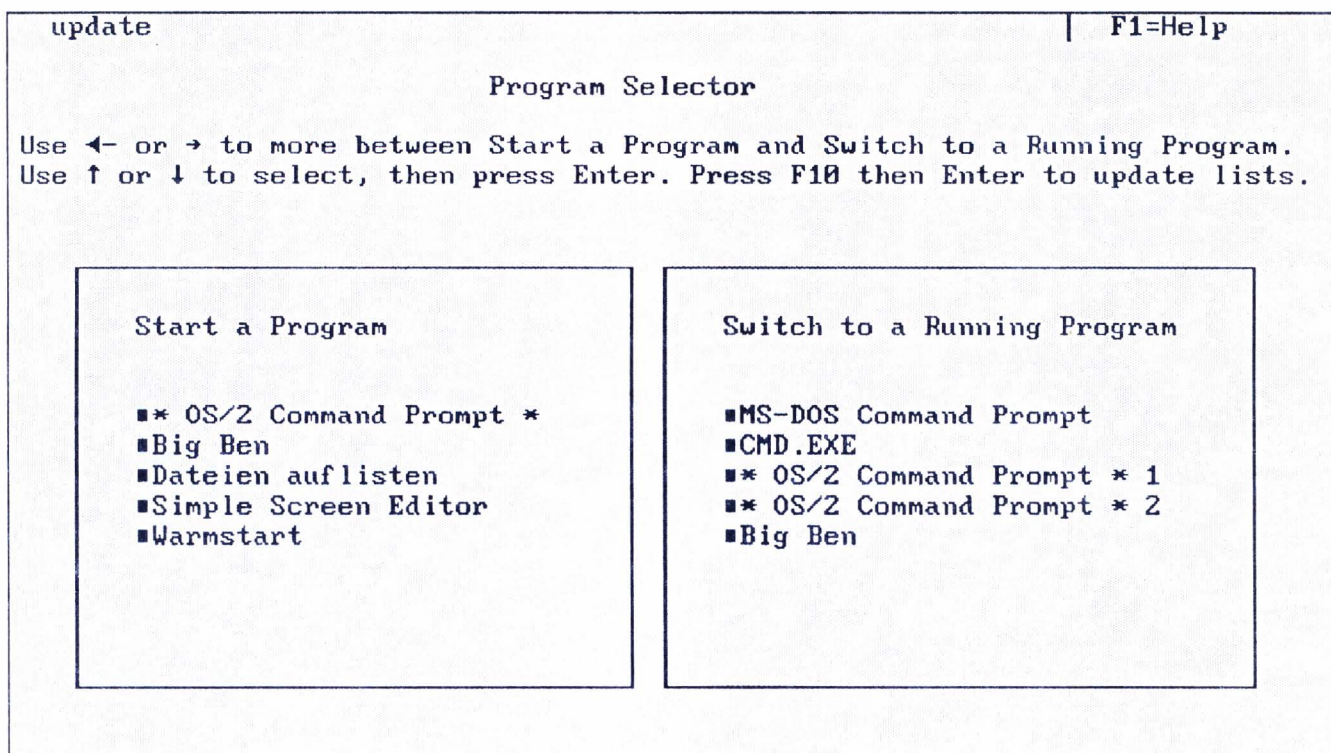


Bild 10.2

```
protshell = c:\os2\shell.exe c:\os2\cmd.exe /k
c:\os2\initenv
```

Wenn Sie Ihren Rechner in MS-OS/2 starten, dann landen Sie direkt im Programm-Selektor, der in diesem Anfangszustand zwei Fenster zeigt. Das sieht dann so aus: (Bild 10.1)

Der Programm-Selektor ist eine Multitasking-Benutzeroberfläche, die in vier Fenstern organisiert ist. Die ersten beiden Fenster enthalten Listen von Programmen. In dem einen können die Programme aufgerufen, das heißt gestartet werden. In dem anderen Fenster kann man schon laufende Programme gezielt auswählen und auf dem Bildschirm sichtbar machen. In diesem Anfangszustand kann entweder zum laufenden Real-Mode, das ist der MS-DOS-Command-Prompt, geschaltet oder ein OS/2-Command-Prompt gestartet werden. Wir werden sehen, daß es in MS-OS/2 mehrere Command-Prompts geben kann, denn das ist ja die Grundvoraussetzung für Multitasking.

Im vollen Multitasking-Betrieb könnte der Programm-Selektor den folgenden Bildschirm zeigen. Fünf Bildschirmgruppen stehen bereit: das OS/2 Prompt, Big Ben, das ist ein kleines Beispielprogramm von Microsoft, das die Uhrzeit generiert, ein Programm, das alle Dateien der Festplatte auflistet, eine kleine Textverarbeitung und ein Programm, das einen Warm-

start produziert, das heißt das den PC neu hochfährt. Das zeigt das linke Fenster.

Im rechten Fenster stehen die gerade laufenden Programme. Das sind erstens der command.com, das Betriebssystem im Real-Mode, zweitens der cmd.exe, das Betriebssystem im Protected-Mode, drittens und viertens je ein OS/2 Prompt - was immer da gerade gemacht wird - und fünftens die Uhr Big Ben (Bild 10.2).

Sie müssen also für den Multitasking-Betrieb zwei wichtige Punkte auseinanderhalten:

1. Die Start-Liste (start list) erlaubt Ihnen, eine neue Bildschirmgruppe zu starten. Diese neue Bildschirmgruppe muß sich allerdings schon in der Liste befinden. Die Liste von Programmen und von Bildschirmgruppen sind in zwei Files (shell11f.aii und shell11f.aif) gespeichert. Das hat den Sinn, die Bildschirmgruppen, die man immer wieder braucht, schon nach dem Booten bereit zu haben. Es ist also zu beachten, daß in der Start-Liste die Programme und Bildschirmgruppen stehen, die noch nicht unbedingt laufen müssen, aber bereit stehen, um aufgerufen zu werden.

2. Dagegen stehen in der Switch-Liste (switch list) die Programme und Bildschirmgruppen, die schon im

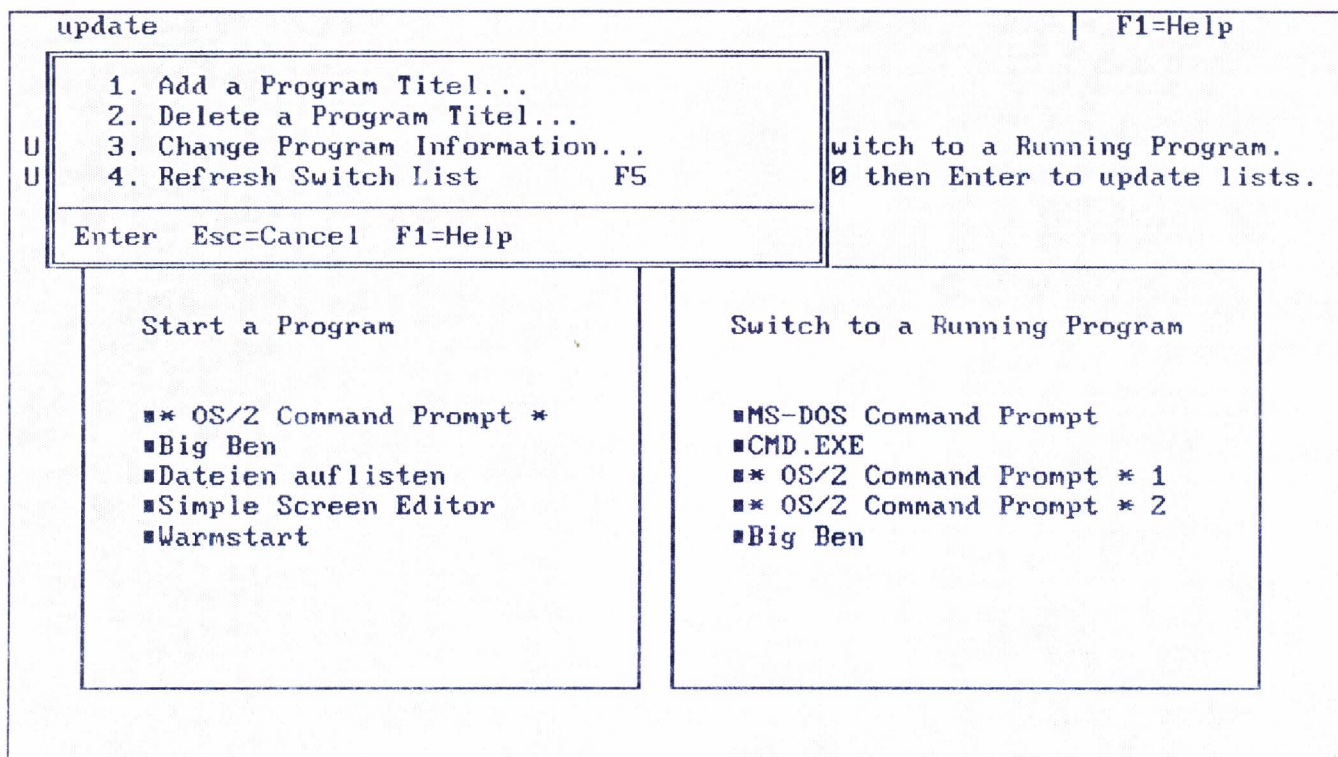


Bild 10.3

Multitasking laufen. Die Switch-Liste erlaubt Ihnen, zu jeder Bildschirmgruppe zu wechseln, die sich in der Liste befindet. Dabei werden die Pfeiltasten zur Wahl und die <Return>-Taste zur Ausführung benutzt. Das so gewählte Programm erscheint auf dem Bildschirm in der Phase, in der sich seine Ausführung gerade befindet. Das Programm, das vorher auf dem Bildschirm war, wird in den Hintergrund gestellt und damit unsichtbar. Die Switch-Liste ermöglicht Ihnen also, einen virtuellen Bildschirm auf den physikalischen umzuschalten und umgekehrt.

Wie bereits angemerkt, läuft MS-OS/2 als Betriebssystem selbst im Multitasking-Mode. Das heißt, daß man gleichzeitig und parallel mehrere Betriebssystem-Befehle ausführen kann:

Formatieren z.B. auf dem einen Prompt (* OS/2 Command Prompt * 1) und Dateien-Drucken auf dem anderen Prompt (* OS/2 Command Prompt * 2).

Nach den bisherigen Beschreibungen ist schon klar, daß der Programm-Selektor noch die Möglichkeit bieten muß, Bildschirmgruppen in die Start-Liste aufzunehmen. Dies und noch einiges mehr leistet das dritte Fenster, das Update-Fenster. Es wird mit der <F10>-Taste von der oberen Bildschirmleiste aus aufgeklappt, und mit der <Enter>- bzw.

<Return>-Taste gelangt man in die Menü-Punkte (Bild 10.3).

"Update" erlaubt, im ersten Menü-Punkt Programme zur Start-Liste hinzuzufügen (additions), im zweiten Menü-Punkt Programme aus der Liste zu löschen (deletions) und im dritten Menü-Punkt zu verändern. Der vierte Menü-Punkt ermöglicht es dem Bediener, die Switch-Liste "aufzufrischen", das heißt die Programme aus der Liste zu entfernen, die schon zuendgekommen sind.

Nach Wahl eines Menü-Punktes wird der Bediener im Dialog unmißverständlich geführt. Nur zum Umgang mit den Punkten eins und drei des Update-Menü seien ein paar Erläuterungen geben. Dort wird der Bediener aufgefordert, einige Eingaben zu machen. Was das für Eingaben sind, erklären wir am Beispiel.

Ziel: Sie wollen eine Bildschirmgruppe einrichten, in der "Microsoft-Multiplan" läuft. Multiplan ist ein Tabellenkalkulationsprogramm, das demnächst als MS-OS/2-Version erhältlich ist. Um ein konkretes Beispiel zu haben, nehmen wir an, wir hätten dieses OS/2-Multiplan schon im Verzeichnis MP auf der Festplatte installiert.

Bild 10.4

Weg: Wir rufen nun im Programm-Selektor das UPDATE Menü auf, indem wir die F10-Taste betätigen, und wählen den ersten Menü-Punkt "Add a Program Title..." aus. Das können wir durch Eingabe des ersten Buchstabens "A" tun oder durch Auswahl mit der Pfeiltaste und anschließender Betätigung der Return-Taste. Es erscheint folgendes Bild: (Bild 10.4)

Zuerst die allgemeinen Erläuterungen:

Die Titel-Variable (Program Title) ist der Name der Bildschirmgruppe, den der Benutzer in der Liste sehen will. Er muß nicht mit dem Programm-Namen übereinstimmen.

Mit Pfad (Program Pathname) ist die volle Kennzeichnung von Laufwerk:Pfad\Programmname für den Aufruf des Programmes bzw. der Anwendung gemeint.

Unter einer Parameterliste (Program Parameters) versteht man alle Eingaben, die das Programm benötigt, um korrekt abzulaufen. Bei der Abfrage sind die folgenden Eingaben möglich:

1) eine Liste der Parameter, sofern sie im Voraus bekannt sind und vom Programm gebraucht werden;

2) ein Fragezeichen (?) in der ersten Spalte verursacht, daß der Benutzer beim Programmaufruf gefragt wird, was er eingeben will;

3) ein Leerzeichen (Blank) bei Programmen ohne Parameter. Big Ben ist zum Beispiel ein Demo-Programm ohne zusätzliche Eingaben (Parameter). Es wird folgendermaßen der Programmliste hinzugefügt:

Program Title: Big Ben

Program Pathname: c:\demos\bigben\bigben.exe

Das Programm bigben.exe liegt in einem eigenen Verzeichnis namens bigben, das seinerseits im Verzeichnis demos liegt, in dem alle Demoprogramme von Microsoft aufbewahrt sind.

Wir kehren zu unserer Aufgabe zurück.

Ziel: Wir schaffen eine Bildschirmgruppe für Multiplan.

Weg: Geben Sie ein, was der folgende Bildschirm zeigt! Bei der Eingabe ist zu beachten, daß man nur mit den Pfeiltasten in die nächsten Eingabe-Felder gehen kann. Erst nach Abschluß der gesamten Eingabe betätigen wir die Return-Taste und gelangen wieder in den Anfangsbildschirm des Programm Se-

Bild 10.5

lectors. Wenn Sie vor Eingabe des gesamten Textes die Return-Taste betätigen, dann springen Sie zu früh aus der Eingabe heraus und haben eine unvollständig beschriebene Bildschirmgruppe. Sie müssen dann mit dem Punkt 3 des Update-Menüs ergänzen. Dort erhalten Sie dasselbe Eingabe-Fenster zum editieren, nachdem Sie die entsprechende Bildschirmgruppe ausgewählt haben (Bild 10.5).

Bei der Eingabe gehen wir davon aus, daß Multiplan wie bisher mit mp aufgerufen wird. Im Feld für Programm-Parameter geben wir ein Leerzeichen (Leertaste) ein. Der einzig mögliche Parameter bei Microsoft-Standard-Software-Programmen ist ein Dateiname - in diesem Falle für eine bestimmte Kalkulationstabelle. Wenn man das so wollte, könnte man diese hier eintragen und landete dann beim Aufruf direkt in seiner konkreten Anwendung. Mit der Eingabe eines Fragezeichens würde man jedesmal zur Eingabe des Dateinamens aufgefordert.

Das vierte und letzte zu beschreibende Fenster ist aufklappbar wie das Update-Fenster. Es enthält kontext-sensitive Hilfen, das sind Hilfen, die von der Situation der Bedienung abhängen. Sie können jederzeit durch Betätigung der <F1>-Taste aufgerufen werden und geben dann Informationen zu der Situation der Bedienung, in der Sie sich gerade befinden. In den Hilfe-Fenstern kann mit den Pfeiltasten geblät-

tert werden, wenn der Text für das Fenster zu groß oder besser: zu lang ist.

Ziel: Nach Aufruf des Update-Fensters wollen Sie die Hilfen einsehen.

Weg: Benutzen Sie die Hilfe-Taste <F1>, um Informationen über das zu erhalten, was Sie gerade in dieser Situation brauchen können.

Sie erhalten zuerst Erläuterungen zum Problem "Add a Program Title". Wenn Sie dann nochmals Hilfe rufen, erscheint der folgende Bildschirm (Bild 10.6).

In der ersten Version von MS-OS/2 für Software-Entwickler gab es noch einen schlichteren Session-Manager. Im Vergleich zu dem vorigen Session-Manager ist der Programm-Selektor - obwohl er noch nicht grafisch, sondern zeichen-orientiert ist - schon weit komfortabler und läßt ahnen, welchen Komfort man für die Bedienung von MS-OS/2 mit dem Windows-Presentation-Manager plant.

Ziel: Wir wollen nun die Bildschirmgruppe "Multiplan" aktivieren, das heißt wir wollen Multiplan starten.

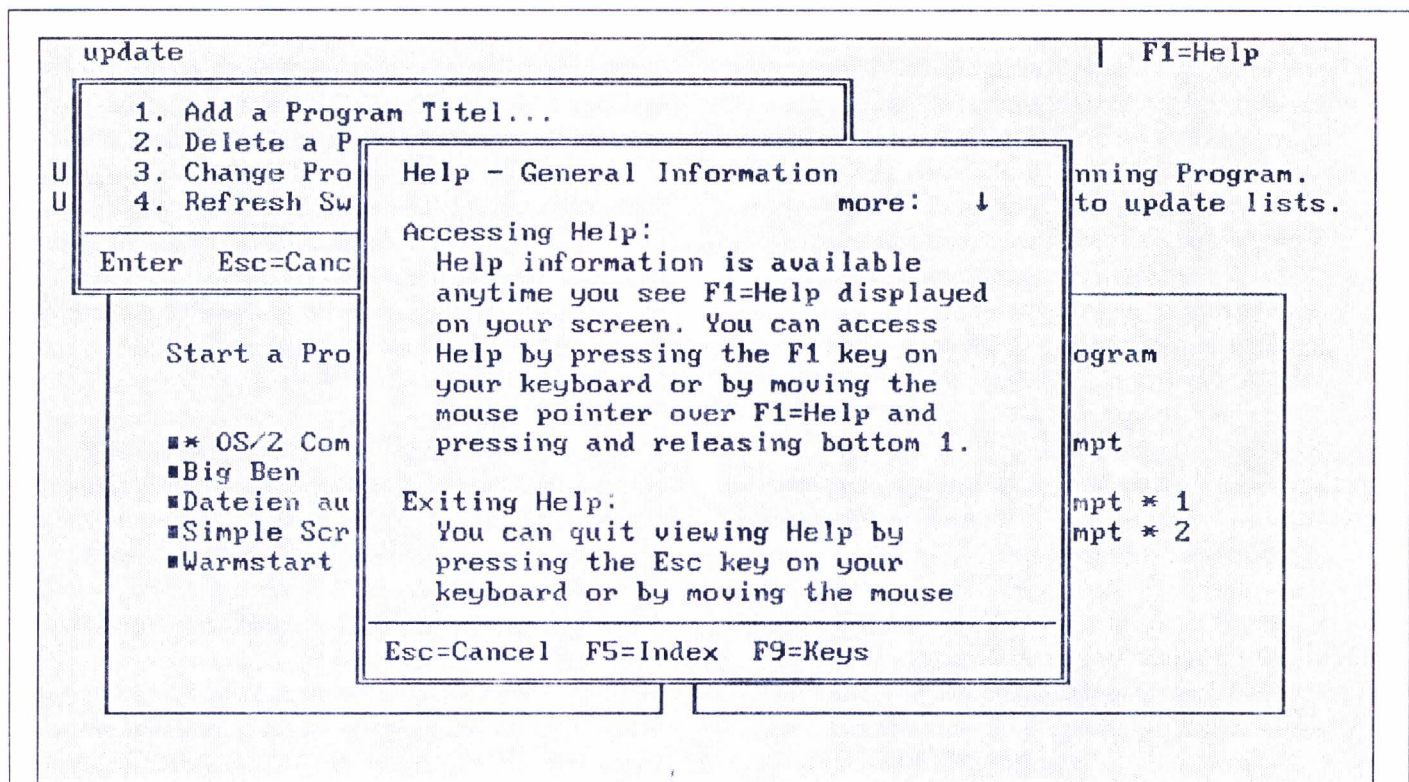


Bild 10.6

Weg: Gehen Sie mit den Pfeiltasten im "Start a Program" Menü auf die Option Multiplan und bestätigen Sie Ihre Wahl mit der Return-Taste.

Resultat: Sie gelangen direkt in die Tabellenkalkulation.

Ziel: Sie wollen wieder zum Programm-Selektor zurück, um auch den Simple-Screen-Editor zu aktivieren.

Weg: Rufen Sie also erneut den Programm-Selektor auf, indem Sie Taste <Strg> niederhalten und die Esc-Taste anschlagen. Wählen Sie dann die Zeile "Simple Screen Editor" mit der Pfeiltaste in der Start-Liste. Schon haben Sie das Gewünschte augenblicklich auf dem Bildschirm.

Ziel: Sie wollen sich, bevor Sie in diesem Editor einen kleinen Text schreiben, in das Programm BIGBEN.EXE einblenden und auf die Uhr sehen. Sie haben zwei Möglichkeiten.

Erster Weg: Rufen Sie den Programm-Selektor mit der Tastenkombination <Strg>-<ESC> auf, wählen Sie in der Switch-Liste Big Ben und schon haben Sie die Uhr auf dem Bildschirm.

Zweiter Weg mit Zusatz-Ziel: Gleichzeitig wollen Sie sich noch einen Überblick über den augenblicklichen Zustand der übrigen Programme verschaffen. In diesem Fall brauchen Sie den Programm-Selektor gar nicht erst aufzurufen, sondern Sie betätigen stattdessen mehrmals die Tastenkombination <Alt>-<ESC>. Dadurch blenden Sie sich nacheinander in die laufenden Programme ein und treffen schließlich auch auf das Programm BIGBEN.EXE.

Wenn Sie also in alle laufenden Programme kurz hineinschauen wollen, um sich über den Stand der Bearbeitung zu informieren, dann ist es ziemlich umständlich, den Sitzungswechsel über den Programm-Selektor durchzuführen. Deshalb ist ein Sitzungswechsel auch direkt, das heißt ohne Rückkehr zum Session-Manager-Menü, möglich. Einen solchen direkten Sitzungswechsel können Sie stets durchführen, indem Sie die Tastenkombination <Alt>-<ESC> betätigen. Bei mehrmaliger Betätigung von <Alt>-<ESC> durchlaufen Sie die Sitzungen zyklisch in der Reihenfolge, wie sie in der Switch-Liste aufgeführt sind.

Wir haben es oben schon erwähnt, daß der Programm-Selektor unter anderem die Betriebsarten auswählen läßt. Wählen Sie MS-DOS Command Prompt in der Switch-Liste, wenn Sie in den Real-Mode wechseln wollen und irgendeinen OS/2 Command

Prompt, wenn Sie in den Protected-Mode wechseln wollen. Sie gelangen natürlich auch während eines zyklischen <Alt>-<ESC>-Durchlaufes in die jeweiligen Betriebsarten. Sie können auch am Programm-Selektor vorbei zwischen REAL-MODE und PROTECTED-MODE mit der Tastenkombination ALT-umschalten. Im Zusammenhang mit mehreren OS/2 Command Prompts im Protected-Mode entsteht häufig folgende Aufgabe:

Ziel: Sie haben auf einem OS/2 Command Prompt eine Diskette formatiert, während Sie auf einem anderen OS/2 Command Prompt Dateien eingesehen haben, und wollen nun die Teilsitzung zur Formatierung beenden. Sie sollten wie folgt vorgehen:

Weg: 1. Begeben Sie sich mit <Strg>-<ESC> in die Switch-Liste des Programm Selectors. 2. Markieren Sie die Teilsitzung, die Sie beenden wollen und betätigen Sie die Eingabetaste. 3. Vergewissern Sie sich, daß der MS-OS/2-Befehl FORMAT in dieser Sitzung beendet ist, da Sie die Sitzung sonst nicht abschließen können. 4. Geben Sie den Befehl EXIT ein und betätigen Sie die Eingabetaste.

Resultat: Nach der Ausführung dieser Aktionen befinden Sie sich wieder im Programm-Selektor, die soeben beendete Sitzung jedoch wird in der Switch-Liste nicht mehr angezeigt.

Bemerkung: Bevor Sie eine komplette Sitzung beenden können, müssen Sie sicherstellen, daß erstens alle MS-OS/2-Befehle und zweitens alle Programme in dieser Sitzung beendet sind. MS-OS/2-Befehle enden im allgemeinen von selbst. Wenn Sie einen MS-OS/2-Befehl dennoch abbrechen wollen, so betätigen Sie einfach die Tastenkombination <Strg>-<C>. Interaktive Programme können Sie durch die Eingabe eines entsprechenden Befehls beenden.

Fassen wir abschließend noch einmal das Wichtigste zusammen: Alles, was man im Ganzen auf den Bildschirm rufen kann, wird in MS-OS/2 Bildschirmgruppe genannt. Das kann eine Familie unterschiedlichster Prozesse sein. Wenn solche Prozesse im Vordergrund ablaufen, dann befinden sie sich auf dem physikalischen Bildschirm. Sie benutzen die Hardware. Sind sie stattdessen im Hintergrund, dann geben diese Prozesse ihre Ausgaben auf einen imaginären Bildschirm, der nur der Möglichkeit nach auf dem physikalischen Bildschirm sichtbar werden kann. Man nennt ihn deshalb virtuellen Bildschirm. Dieses Konzept macht es aus, daß man augenblicklich sieht, was gerade passiert, wenn man mit dem Session-Manager einen Prozeß aus dem Hintergrund nach "vorne" holt.

11 MS-DOS und MS-OS/2 sind einander sehr ähnlich: 90 % der Befehle sind identisch

Man unterscheidet sowohl in MS-DOS als auch in MS-OS/2 zwei Befehlstypen: Interne Befehle und externe Befehle. Interne Befehle sind Befehle, die unmittelbar durch den Befehlsinterpreter abgearbeitet werden. Diese Befehle sind feste Bestandteile der Interpreter und erscheinen daher auch nicht im Verzeichnis der MS-OS/2-Systemdateien. Die internen Befehle für den Real-Mode befinden sich also innerhalb der Datei command.com und die internen Befehle für den Protected-Mode sind in den cmd.exe integriert.

Interne Befehle können von MS-OS/2 sofort ausgeführt werden, da sich die jeweiligen Befehlsinterpreter permanent im Arbeitsspeicher befinden. Wichtig

ist es, die Unterscheidung zu beachten, wenn Sie auf Ihrer Festplatte mit wechselnden Pfaden arbeiten. Die externen Befehle hält man normalerweise in einem Unterverzeichnis. Liegt dieses Unterverzeichnis nicht im Suchpfad, dann können die externen Befehle auch nicht aufgerufen werden. Demgegenüber sind die internen Befehle "pfad-unabhängig".

Die vollständige Liste der internen Befehle umfaßt folgende MS-OS/2-Kommandos:

break, chcp, chdir, cls, copy, date, del, detach, dir, dpath, echo, endlocal, extproc, exit, for, goto, if, mkdir, path, pause, prompt, rem, ren, rmdir, set, setlocal, shift, time, type, ver, verify, vol.

In dieser Liste sind alle speziellen sogenannten Batch-Befehle enthalten. Eine Batch-Datei - auch Stapeldatei oder Stapel-Programm genannt - ist eine Abfolge von Betriebssystembefehlen, die einer nach dem anderen wie von einem Stapel geholt und abgearbeitet werden. Batch-Befehle können nur in solchen Programmen vorkommen, also nicht interaktiv auf dem OS/2-Prompt. Alle diese Batch-Befehle sind nicht nur für Real-Mode und Protected-Mode dieselben, sondern auch schon MS-DOS bekannt (vgl. Tabelle 11.1)

Von allen 32 internen Befehlen sind nur 3 der "normalen" Befehle neu: chcp, detach und dpath. Das sind ungefähr 10% - also 90% bleibt beim Alten! Drei weitere Batch-Befehle für den ausgefallenen Bedarf kommen hinzu: endlocal, extproc und setlocal.

Externe Befehle können gewissermaßen nicht vollständig angegeben werden, denn jeder Benutzer kann sich neue MS-OS/2-Befehle dieser Art schreiben. Jeder Dateiname mit einer der Erweiterungen .com, .exe, .cmd oder .bat wird als externer Befehl betrachtet. So sind zum Beispiel format.exe oder diskcopy.exe externe Befehle. Da alle externen Befehle eigenständige Dateien sind, können Sie sich auch selbst MS-OS/2-Befehle, sogenannte Utilities (Hilfsprogramme), erstellen. Dazu braucht man kein Programmierer zu sein. Wenn Sie z.B. eine Autoexec.bat schreiben, dann erzeugen Sie einen externen Befehl.

Wenn Sie einen externen Befehl aufrufen, brauchen Sie die Dateierweiterung nicht einzugeben. MS-OS/2 sucht dann von sich aus nach Dateien mit der Namens-Erweiterung .com, .exe, .cmd (oder .bat). Wenn Sie mehrere externe Befehle haben, die bis auf die Erweiterung den gleichen Dateinamen besitzen,

dann sucht MS-OS/2 die Datei in der Reihenfolge, die wir angegeben haben. Und der Befehl, der zuerst gefunden wird, kann ausgeführt werden. Aber das war auch schon bei MS-DOS so der Fall - mit der einzigen Ausnahme, daß es dort noch keine Stapelprogramme mit der Erweiterung .cmd gegeben hat (vgl. die Abschnitte 7 und 8).

Die Liste der externen Befehle, wie sie von Microsoft ausgeliefert werden, umfaßt die folgenden Kommandos:

ansi, append, assign, attrib, backup, chkdsk, cmd, command, comp, diskcomp, diskcopy, fdisk, find, format, graftabl, helpmsg, join, keyb, label, mode, more, patch, print, recover, replace, restore, sort, spool, subst, sys, tree, xcopy.

Von allen 32 externen Befehlen sind nur 4 neu: cmd, helpmsg, patch und spool. In summa heißt dies ca. 90% ist beim Alten geblieben.

Abschließend wollen wir die verschiedenen Merkmale der Befehle in einer vollständigen Tabelle zusammenfassen. Die Spaltenüberschriften sind folgendermaßen zu verstehen:

In der Spalte Befehl ist der Name des Befehls aufgeführt, der beschrieben werden soll. Ein Stern in der Real-Spalte bedeutet, daß dieser Befehl nur im Real-Mode verwendet werden kann. Entsprechendes gilt für die Protect-Spalte. Befehle in der Real+Pro-Spalte können mit derselben Syntax in beiden Modi verwendet werden. Kommen Sternchen sowohl in der Real- als auch in der Protect-Spalte vor, dann ist der Befehl in beiden Betriebsarten verwendbar, er besitzt aber eine jeweils leicht unterschiedliche Syntax. Die

Die Batch-Befehle

Befehl	Real	Protect	Real+Pro	Intern	Extern	Netzwerk
call			*	*		
echo			*	*		
endlocal neu			*	*		
extproc neu			*	*		
for			*	*		
goto			*	*		
if			*	*		
pause			*	*		
rem			*	*		
setlocal neu			*	*		
shift			*	*		

Tabelle 11.1

Befehlsübersicht

Befehl		Real	Protect	Real+Pro	Intern	Extern	Netzwerk
ansi			*			*	
append		*				*	
assign		*				*	
attrib				*		*	
backup				*		*	
break		*			*		
chcp	neu			*	*		
chdir				*	*		
chkdsk				*		*	-
cls				*	*		
cmd	neu		*			*	
command		*				*	
comp				*		*	
copy				*	*		
date				*	*		
del		*	*		*		
detach	neu		*		*		
dir		*	*		*		
diskcomp				*		*	-
diskcopy				*		*	-
dpath	neu		*		*		
exit				*	*		
fdisk				*		*	-
find				*		*	
format				*		*	-
graftabl		*				*	
helpmsg	neu			*		*	
join		*				*	
keyb			*			*	
label				*		*	-
mkdir		*	*		*		
mode				*		*	
more				*		*	
patch	neu			*		*	
path				*	*		
print				*		*	
prompt				*	*		
recover				*		*	-
ren				*	*		
replace				*		*	
restore				*		*	
rmdir		*	*		*		
set				*	*		
sort				*		*	
spool	neu			*		*	
subst				*		*	-
sys				*		*	-
time				*	*		
tree				*		*	
type		*	*		*		
ver				*	*		
verify				*	*		
vol		*	*		*		
xcopy				*		*	

Tabelle 11.2

Spalten Intern und Extern verstehen sich von selbst, und für den diesbezüglich Interessierten ist noch vollständigkeitshalber die Spalte Netzwerk angefügt. Die mit dem Minuszeichen (-) markierten Befehle sind

nicht netzwerkfähig. Die Befehle, die gegenüber MS-DOS neu sind, haben hinter dem Namen einen entsprechenden Eintrag (Tabelle 11.2).

12 Die Unterschiede bei den Betriebssystembefehlen: Welche neuen Befehle sind zu beachten?

Die Befehle, die nun aufgeführt werden, sind allesamt Protected-Mode-Befehle. Einige davon können auch im Real-Mode verwendet werden. Sie kommen in diese Auswahl, weil sie für MS-OS/2 neu geschaffen wurden. Sie sind teilweise sehr speziell und brauchen den normalen Bediener nicht zu interessieren. Aber es mag trotzdem von Interesse sein, zu sehen, was es unter MS-OS/2 Neues gibt. Und für den fortgeschrittenen Kenner von MS-DOS ist der folgende Überblick sicherlich nützlich. Wir listen die neuen Befehle nach ihrem Ziel, nach ihrem Weg und nach ihrer Syntax nur ganz schlicht auf.

Ziel: ANSI macht die ANSI-Escape-Sequenzen, das sind Steuerzeichen für den Bildschirm, im Protected-Mode verfügbar. Zum Beispiel:

Weg:

ansi on oder ansi off

gibt den Zugriff auf die ANSI-Escape-Sequenzen frei oder sperrt ihn.

Bemerkung: Falls ansi auf OFF gesetzt wird, stehen die ANSI-Escape-Sequenzen nicht mehr zur Verfügung. Das System setzt ansi standardmäßig auf ON. Die Eingabe von ansi ohne Optionen liefert den momentan eingestellten Wert zurück. Den ansi-Befehl gab es auch schon unter MS-DOS, er ist aber hier wesentlich verändert.

Ziel: CHCP zeigt an oder ändert die aktuelle Code-Seite für den Kommandoprozessor cmd.exe oder command.com.

Weg: chcp [nnn]

Bemerkung: nnn ist die Nummer der Code-Seite, die von nun an verwendet werden soll. Code-Seiten sind Übersetzungstabellen für Buchstaben/Zeichen, kurz: für das jeweils gültige Alphabet. Das System besitzt zwei vorbereitete Code-Seiten, von denen durch das chcp-Kommando eine ausgewählt werden kann. Wenn chcp ohne Code-Seite eingegeben wird, werden die aktive und die vorbereiteten Code-Seiten für den Kommandoprozessor der aktiven Bildschirm-Gruppe angezeigt. Es kann jede der vom System vorbereiteten Code-Seiten gewählt werden, die durch das codepage-Kommando in der Datei config.sys definiert sind.

Gültige Code-Seiten sind:

437 für USA
850 für Mehrsprachig
860 für Portugiesisch
863 für Franz.-Kanadisch
865 für Nordisch

Ziel: CMD startet den Kommandoprozessor des Protected-Mode.

Weg: cmd [d:] [Pfad] [/c:nnnnn] [/p] [/c Zeichenkette]

Bemerkung: Dieses Kommando startet einen neuen Kommandoprozessor im Protected-Mode (Der Kommandoprozessor ist ein OS/2-Programm, das alle internen Kommandos beinhaltet).

Beim Start eines neuen Kommandoprozessors wird auch eine neue Kommandoumgebung erzeugt. Diese neue Umgebung ist identisch mit der alten, kann aber modifiziert werden, ohne daß die alte Umgebung beeinträchtigt wird. Außerdem kann mit dem Parameter /e die Größe der neuen Umgebung verändert werden.

Der Kommandoprozessor wird in zwei Teilen in den Hauptspeicher geladen: Der eine ist flüchtig und der andere resident. Der flüchtige Teil wird von manchen Anwendungen überschrieben und muß dann bei Bedarf durch den residenten Teil des Kommandoprozessors vom Laufwerk nachgeladen werden. Die Angabe von Laufwerk und Pfad beschreibt den Zugriffspfad zur Datei cmd.exe, falls der flüchtige Teil des Kommandoprozessors neu geladen werden muß.

Der Parameter /e definiert die Umgebungsgröße, wobei nnnnn eine Größenangabe in Bytes ist, die zwischen 128 und 32768 Bytes liegen muß. Wird keine Angabe gemacht, so werden 128 Bytes angenommen. Liegt nnnn außerhalb des zulässigen Bereichs, so wird eine Fehlermeldung ausgegeben und der Wert auf 128 Bytes angehoben, falls er zu klein war bzw. auf 32768 Bytes gesetzt, falls er zu groß war.

Der Parameter /p weist cmd.exe an, nicht automatisch auf eine höhere Kommandoebene zurückzukehren.

Der Parameter /c muß, falls er benutzt wird, der letzte Parameter im Kommando sein. Er weist den Kommandoprozessor an, die durch die Zeichenkette spezifizierten Kommandos auszuführen und anschließend automatisch auf die höhere Kommandoebene zurückzukehren.

Nachdem ein OS/2-Kommando eingegeben wurde, welches den residenten Umfang von OS/2 vergrößert, ist eine Vergrößerung der Umgebung nicht mehr möglich. Die Größe der Umgebung muß also vor einem solchen Kommando festgelegt werden.

Ziel: DETACH startet einen Hintergrundprozeß.

Weg: detach Kommando (bzw. Programmname)

Bemerkung: Mit dem detach-Kommando ist es möglich, bestimmte Kommandos im Hintergrund ausführen zu lassen. Das bedeutet, daß OS/2

einen unabhängigen Prozeß startet,

die Prozeßkennnummer des neuen Prozesses anzeigt

und sofort zur Eingabe des nächsten Kommandos bereit ist.

Das angegebene Kommando muß für den Protected-Mode zugelassen sein und darf keine weiteren Tastatureingaben benötigen. Ansonsten können Dateien zerstört werden und wichtige Daten verloren gehen.

Ein mit einem detach-Kommando gestarteter Hintergrundprozeß muß von selbst terminieren, da er nicht mehr gestoppt werden kann.

Ziel: DPATH gibt an, in welchen Verzeichnissen eine Anwendung Dateien suchen soll, welche Eingabedaten enthalten, das heißt Dateien, die andere Erweiterungen als .exe, .com, .bat oder .cmd besitzen.

Weg: Um Verzeichnisse anzugeben, die durchsucht werden sollen:

dpath [d:] Pfad [[: [d:] Pfad] ...]

Weg: Um die Pfade zu löschen:

dpath ;

Weg: Um die Pfade anzuzeigen:

dpath

Bemerkung: Das dpath-Kommando teilt einer Anwendung mit, in welchen Verzeichnissen nach Eingabedaten gesucht werden soll, falls diese Eingabedaten nicht im aktuellen oder im angegebenen Verzeichnis gefunden werden.

Das append-Kommando erfüllt im Real-Mode eine ähnliche Funktion wie dpath im Protected-Mode.

Im Gegensatz zu dpath definiert das path-Kommando Suchpfade für externe Kommandos.

Wird von einer einzelnen Anwendung aus ein neuer Kommandoprozessor gestartet, so werden die dpath-Zuordnungen auf den neuen Prozeß vererbt.

Ziel: EXIT verläßt den jeweiligen Kommandoprozessor und kehrt auf die vorhergehende Ebene zurück, falls diese existiert.

Weg: exit

Bemerkung: Werden die Kommandos cmd oder command benutzt, um einen neuen Kommandoprozessor zu starten, so kann das exit-Kommando ver-

wendet werden, um zu dem alten Kommandoprozessor zurückzukehren. Außerdem erlaubt exit den Ausstieg aus einem Anwendungsprogramm zum Kommandoprozessor und die anschließende Rückkehr zum Programm. Das exit-Kommando bleibt ohne Wirkung, falls es keinen Kommandoprozessor gibt, der sich auf einer höheren Ebene befindet.

Ziel: HELPMMSG liefert Informationen über eine Warnung oder Fehlermeldung des Systems.

Weg: helpmsg Meldungskennung

Bemerkung: Jede OS/2-Meldung besitzt eine Meldungskennung der Form "DOSnnnn", wobei nnnn eine vierstellige Meldungsnummer ist. Um mehr Informationen über eine OS/2-Meldung zu erhalten, ist das Kommando helpmsg gefolgt von der Meldungskennung einzugeben.

Als Antwort wird angezeigt:

die Meldung selbst, wobei die darin enthaltenen Variablen von OS/2 durch "****" ersetzt werden können

Erläuterungen zur Bedeutung der Meldung

ein Vorschlag, wie auf die Meldung reagiert werden soll.

Ziel: KEYB spezifiziert ein nationales Tastaturformat.

Weg: keyb yy

Beispiel: Um eine deutsche Tastatur zu benutzen, muß

keyb gr

einggegeben werden.

Bemerkung: yy ist ein Landesschlüssel, der aus zwei Buchstaben besteht. Folgende Landesschlüssel stehen zur Verfügung:

us für USA
uk für Großbritannien
gr für Deutschland
fr für Frankreich
it für Italien
sp für Spanien

Ziel: PATCH ermöglicht die direkte Änderung von Programmcode.

Weg: patch [[d:] Pfadname] [/a]

Bemerkung: Mit dem patch-Kommando lassen sich Änderungen im Programmcode durchführen. Die Bytes des Programmcodes können direkt modifiziert werden oder es können Bytes am Dateiende angefügt werden. Jede Datei, für die ein schreibender Zugriff erlaubt ist, kann durch ein patch-Kommando geändert werden.

Das patch-Kommando besitzt einen automatischen und einen interaktiven Operationsmodus.

Mit dem Parameter /a wird der automatische Operationsmodus ausgewählt. Das patch-Kommando läuft dann über einen vordefinierten Laufwerks- und Pfadnamen, der die Änderungsbefehle für die automatisch zu ändernde Datei oder auch mehrere Dateien enthält.

Im Normalfall wird der interaktive Operationsmodus aktiviert. In der Kommandozeile wird der Name der zu ändernden Datei angegeben. Anschließend wird der Offset, bei dem die Änderungen durchzuführen sind, und die Änderungsdaten angefordert.

Sowohl Offset als auch Daten sind in hexadezimaler Notation einzugeben.

Nach der Angabe des Offset werden die 16 Bytes der Datei bei diesem Offset angezeigt. Die angezeigten Bytes kann man beliebig ändern. Sollen keine Änderungen durchgeführt werden, so ist nur die ESC-Taste zu betätigen.

Der Cursor befindet sich zunächst auf dem ersten Byte. Dieses Byte kann nun durch die Eingabe von einer oder zwei Hexadezimalziffern überschrieben werden. Soll das erste Byte unverändert bleiben, so kann durch die Betätigung der Leertaste der Cursor jeweils zum nächsten Byte bewegt werden. Mit der BACKSPACE-Taste wird der Cursor zurückbewegt, falls ein Fehler gemacht wird. Wird der Cursor hinter das 16. Byte bewegt, so werden die nächsten 16 Bytes angezeigt.

Durch die Eingabe von RETURN wird ein Änderungsvorgang abgeschlossen und die Änderungen zwischengespeichert. Danach wird gefragt, ob weitere

Neue Befehle

Änderungen durchgeführt werden sollen. Wenn ja, wird erneut die Angabe eines Offset verlangt.

Nachdem alle Änderungen eingegeben und zwischengespeichert worden sind, wird gefragt, ob die Änderungen an der Originaldatei durchgeführt werden sollen. Wenn ja, werden die Änderungen in der Reihenfolge, in der sie eingegeben worden sind, durchgeführt.

Warnung: Das patch-Kommando sollte nur verwendet werden, wenn die Auswirkung der Änderungen genau bekannt ist. Bevor ein patch-Kommando eingegeben wird sollte von der zu ändernden Datei eine Sicherheitskopie erstellt werden.

SPOOL startet den Druckerspooler. Neue Kommandos oder Programme können gestartet werden, während im Hintergrund gedruckt wird.

Weg: `spool [d:][Pfad] [/d:Gerät1] [/o:Gerät2]`

Bemerkung: Der angegebene Pfad bezeichnet das Verzeichnis, in dem die temporären Dateien erzeugt werden, die für den Ausdruck bestimmt sind. Wenn kein Pfad angegeben wird, verwendet OS/2 das Verzeichnis `\spool`.

Mit dem `spool`-Kommando kann mehr als ein Druckauftrag abgesetzt werden, während weiterhin Programme oder OS/2-Kommandos ausgeführt werden können. Die Druckdaten werden zwischengespeichert und die Druckaufträge in einer Warteschlange verwaltet, um eine saubere Trennung der Aufträge zu ermöglichen. Wenn ein Anwendungsprogramm gerade die zu druckende Datei bearbeitet, wird die Datei in einer getrennten temporären Datei im Spool-Unterverzeichnis gespeichert. Sobald das Anwendungsprogramm beendet ist und die Datei gedruckt werden

darf, wird die Datei in die Druckerwarteschlange gesetzt und, sowie der Drucker verfügbar ist, ausgedruckt.

Die Parameter des `spool`-Kommandos haben folgende Bedeutung:

`/d:Gerät1`

bezeichnet den Port, an den der Drucker angeschlossen ist. Spool arbeitet sowohl mit parallelen (PRN, LPT1, LPT2, LPT3) als auch mit seriellen Ports (COMx, wobei x eine Zahl von 1 bis 8 ist). Der Standardwert ist LPT1.

`/o:Gerät`

bezeichnet das Ausgabegerät. Wird keine Angabe dazu gemacht, so wird Gerät1 angenommen.

Um die Spool-Routine zu benutzen muß im `print`-Kommando einer der Parameter `/t` oder `/c` angegeben werden.

Sollen Dateien in ein getrenntes Spool-Verzeichnis transferiert werden, so ist zuerst das Verzeichnis mit dem `mkdir`-Kommando anzulegen.

Bevor ein serieller Port im Protected-Mode benutzt werden kann, muß der Gerätetreiber installiert werden.

Beispiel: Das folgende Kommando veranlaßt, daß Druckaufträge in das Verzeichnis `\drucker` transferiert und auf einem seriellen Drucker ausgegeben werden, der an den Port COM1 angeschlossen ist:

```
spool \drucker /d:COM1
```


13 So läuft MS-DOS-Software unter MS-OS/2: Im Real-Mode und verändert im Protected-Mode

Der Compatibility-Mode in MS-OS/2

Um die Kompatibilität zu existierenden Anwendungen unter MS-DOS 3.x zu erhalten, stellt MS-OS/2, wie wir bereits wissen, eine Umgebung bereit, in der solche Anwendungen ablaufen können. Diese Anwendungen werden im Real-Mode des Prozessors 80286 und MS-OS/2 abgewickelt.

Anwender und Programmierer fragen sich mit recht, ob ihre Anwendungsprogramme, die sie in der Standard-Software angelegt haben, überhaupt noch unter MS-OS/2 laufen können. Die Antwort ist positiv. Nahezu alle bislang unter MS-DOS geschriebenen Programme laufen auch unter dem Betriebssystem MS-OS/2. Nicht unter einem dritten Zustand des Prozessors, sondern schlichtweg unter dem Real-Mode wie bisher. Die Umgebung der Kompatibilitätsbox oder des Compatibility-Mode - wie der Real-Mode auch genannt wird - entspricht einer DOS 3.x-Umgebung.

Die meisten Standard-Software-Programme arbeiten in dieser Umgebung problemlos. Vermutlich nur 5% der Anwender-Software wird anzupassen sein. Es handelt sich dabei um Programme, die nicht-dokumentierte Dienste unter MS-DOS 3.x in Anspruch nehmen. MS-OS/2 kennt die meisten dieser undokumentierten Dienste insbesondere auf den Interrupt 21h nicht. Ein Interrupt ist eine Programmunterbrechung, in der Regel um Ein- und Ausgaben (I/O) zu steuern. Die dokumentierten DOS-Dienste dagegen existieren alle für den Compatibility-Mode weiter.

Anwender-Programme können jeden Hardware-Interrupt aufrufen - mit Ausnahme des Uhrentaktes (CMOS clock/calendar interrupt) und mit Ausnahme der Interrupts, die von MS-OS/2-Geräte-Treibern schon benutzt werden. Aber das sind Dinge, die den Anwender nicht zu interessieren brauchen.

Es wird in Zukunft eine ganze Zeit lang mit vier Arten von Programmen fest zu rechnen sein. Das sind einmal die gewohnten und altbekannten MS-DOS-Programme. Dazu treten Programme, die im kompatiblen Real-Mode unter dessen Befehlsinterpreter Command.com laufen werden. Diese beiden Programmgruppen haben vieles gemeinsam. Die dritte Gruppe bilden die neuen Programme, die auf dem Befehlsinterpreter cmd.exe aufsitzen, deshalb auch nur im Protected-Mode laufen und alle Merkmale dieser neuen Betriebsart tragen. Eine vierte Art von Programmen wird in der Lage sein, in beiden Betriebsarten lauffähig zu sein - und zwar nicht deshalb, weil sich solche Programme in zwei Versionen präsentieren, sondern in ein und derselben Version. Das gelingt dann, wenn die Software-Entwickler beim Programmieren nur Betriebssystemaufrufe aus dem sogenannten Family Application Program Interface (Schnittstelle für Anwendungsprogramme aus der MS-DOS/MS-OS/2-Familie) benutzt haben.

API ist ein mittlerweile geläufiger Ausdruck für Programmier-Hilfen, die dem Programmierer von Seiten einer anderen Software zur Verfügung gestellt werden. Zumeist handelt es sich um Hilfen des Betriebssystems oder sehr hardwarenaher Rahmenprogramme. So kennt zum Beispiel Windows oder der MS-OS/2-Presentation-Manager jeweils auch ein API. Das FAPI ist eine Bibliothek von 91 verschiedenen Funktionen, die zugleich Bestandteil der Systembibliothek API von MS-OS/2 sind. Diese Funktionen sind Calls (Aufrufe), mit denen Programmierer direkt auf den Betriebssystemkern zugreifen können. Sie ersetzen also den traditionellen MS-DOS-Interruptzugang und gewährleisten zugleich, daß der Zugriff auf die Hardware vom Betriebssystem kontrolliert werden kann. Dies ist gerade in einer Multi-tasking-Umgebung von besonderer Bedeutung.

MS-DOS Programme im Real-Mode

Wir haben oben einige allgemeine Anmerkungen zu den Umständen gemacht, die die Lauffähigkeit von

MS-DOS-Programmen in der Kompatibilitätsbox behindern. Für den Bediener am Wichtigsten allerdings ist vermutlich die Einschränkung, daß im Real-Mode - auch in einer optimalen Hardware-Konfiguration, das heißt wenn das Maximum von 640 Kilobyte in der Kompatibilitätsbox zur Verfügung steht - nur Programme laufen, die nicht mehr als 540 Kilobyte Arbeitsspeicherplatz benötigen. Denn wie bei den allgemeinen Hardware-Voraussetzungen schon einmal angemerkt, benötigt das Betriebssystem im Real-Mode für sich selbst 100 Kilobyte Arbeitsspeicherplatz. Das ergibt dann als Rest gerade die 540 Kilobyte freien Arbeitsspeicher. Diese Voraussetzung für die Lauffähigkeit eines Programmes in der Kompatibilitätsbox kann der Bediener leicht selbst überprüfen, denn der Speicherplatzbedarf ist bei jedem Software-Produkt angegeben.

Vom Autor selbst getestet - wie ausreichend auch immer - wurden folgende Programme: Microsoft-Word, Wordstar in der ersten Fassung für den 16 Bit-Rechner, Microsoft-Multiplan, Microsoft-Chart, Microsoft-Rbase-System. Gerade für das letztgenannte Programm Rbase-System war es besonders wichtig, einen voll ausgebauten Arbeitsspeicher für den Real-Mode zur Verfügung zu haben. Denn Rbase-System braucht für sich selbst ca. 480 Kilobyte.

Die hier aufgeführte Liste von Programmen ist sicherlich nicht vollständig. Sie sind aber Beleg dafür, daß wirklich weitverbreitete Software in der Kompatibilitätsbox problemlos zu benutzen ist.

Der Bediener muß keine Befürchtungen haben, daß er mit seinen gewohnten Programmen und mit seinen Daten nicht mehr arbeiten kann, wenn er nunmehr einen Rechner mit dem neuen Betriebssystem MS-OS/2 anschafft.

Neben den Standard-Software-Paketen, die natürlich für die meisten Bediener den bedeutendsten Stellenwert einnehmen, hat Microsoft nun auch den vollen Satz seiner Programmiersprachen für MS-OS/2 im Angebot. Diese Programmiersprachen wie BASIC, Pascal usw. können wahlweise im Real oder Protected-Mode verwendet werden.

Über den Aufruf von traditionellen MS-DOS-Programmen in der Kompatibilitätsbox sind alle wichtigen Angaben im Abschnitt über den Programm-Selektor gemacht worden. Es sei hier nur um der Geschlossenheit der Information willen noch einmal

kurz zusammengefaßt. Sie rufen im Programm-Selektor in der Switch-Liste den MS-DOS-Command-Prompt auf und gelangen so in den Real-Mode von MS-OS/2. Dort geben Sie wie bisher Ihren Programmnamen bzw. Programmaufruf über die Tastatur ein und bestätigen mit Return. Zum Beispiel:

C:\>MP (Multiplan-Aufruf)

Sie erhalten die übliche Programm-Oberfläche von Multiplan und beginnen zu arbeiten, erstellen Ihre Kalkulationstabelle, beenden irgendwann, wenn Sie fertig sind, Ihre Arbeit und können dann entweder das nächste Programm aufrufen oder in den Protected-Mode von MS-OS/2 wechseln (switchen).

Sie können aber auch mitten aus der laufenden Bedienung von Multiplan herausspringen. Betätigen Sie die Steuerung-Escape-Tastenkombination, dann wird der Programm-Selektor aufgerufen und Sie können ohne weiteres unter dem Protected-Mode eine Bildschirmgruppe laufen lassen, die Ihnen zur Verfügung steht, oder unter dem MS-OS/2 Command Prompt ein Multitasking-Programm laufen lassen.

Wenn Sie ein Programm im MS-DOS-Compatibility-Mode (Real-Mode) laufen haben und Sie wechseln währenddessen in den Protected-Mode, dann wird es sofort unterbrochen und in den Hintergrund gestellt. Es ist dann gleichsam eingefroren. Im Hintergrund eingefrorene Programme erhalten keine CPU-Zeit (Rechenzeit) mehr und auch keine Hardware-Interrupts. Das kann dann zu Problemen führen, wenn das abgestellte Programm zum Beispiel auf fortwährende Zeit-Interrupts angewiesen ist. Das kommt zwar selten vor, aber man sollte dies dennoch für den Ausnahmefall wissen.

Von MS-DOS her bekannte Programme im Protected-Mode:

Außer einigen Beispielprogrammen, die vor allem Software-Entwickler zur Demonstration von Multitasking-Funktionen im sogenannten SDK (Software Development Kit) mitgeliefert bekommen, konnten bis zum Redaktionsschluß keine echten, voll funktionsfähigen Anwendungsprogramme getestet werden. Microsoft führt folgende Programme für MS-OS/2 auf:

Platinum (von Advanced Business Microsystems Inc.), Informix-4GL (von Informix Software Inc.), Informix-SQL (von Informix Software Inc.), Informix-

ESQL/C (von Informix Software Inc.), C-ISASM (von MicroPro International), Rbase for OS/2 (von Microrim Inc.), ORACLE für OS/2 (von Oracle Corporation), Bridge/386 (von Softbridge Microsystems Corp.), Bridge/286 (von Softbridge Microsystems Corp.), Bridge/OSSoftbridge Microsystems Corp., WordPerfect (von WordPerfect Corporation), PlanPerfect (von WordPerfect Corporation), DataPerfect (von WordPerfect Corporation), WordPerfect Office (von WordPerfect Corporation), DBMS und Networking Software (von WordTec Systems, Inc.).

Von Microsoft angekündigt: Microsoft-Multiplan für MS-OS/2, Microsoft-Word und die gesamte Standard-Software-Produktpalette.

Daß es Multitasking-Versionen zu den Programmiersprachen gibt, wurde oben schon erwähnt. Gemäß den Ankündigungen werden die Microsoft-Produkte in einem Paket als zwei Versionen ausgeliefert, einmal für MS-DOS und einmal für MS-OS/2. Damit soll demonstriert werden, daß Microsoft auch in Zukunft beide Betriebssystem-Welten unterstützen will und unterstützt wird: Die "alte" Welt von MS-DOS und die "neue" Welt von MS-OS/2.

An Multitasking-Funktionen soll Multiplan das parallele Arbeiten in mehreren Spread-Sheets (Kalkulationstabellen), aber auch den parallelen Ablauf von verschiedenen Prozessen über dasselbe Spread-Sheet bieten. Man wird dann vermutlich zum Beispiel ein komplexeres Multiplan-Makro zur Konsolidierung von Daten über das Spread Sheet zur gleichen Zeit

laufen lassen können, zu der man auch Daten eingibt oder etwa bestimmte Formatierungen von Spalten vornimmt.

Für Microsoft-Word sind die folgenden Multitasking-Funktionen für MS-OS/2 angekündigt worden: Man kann im Hintergrund Dokumente ausdrucken und gleichzeitig auf dem Bildschirm ein anderes Dokument bearbeiten. Man wird aber auch im Hintergrund langdauernde Such- oder Wechselläufe in langen Texten verrichten lassen können, während man im Vordergrund am Bildschirm weiterarbeitet. Für solche langdauernden Hintergrundprozesse lassen sich noch viele Beispiele finden: die Erstellung eines Indexes über einen Text, die Erstellung des Inhaltsverzeichnisses über einen Text, die Prüfung der Rechtschreibung oder das Heraussuchen von Namen und Verfasserzeichen über den Text-Manager aus einer langen Liste von unterschiedlichen Dokumenten.

Mehr kann an dieser Stelle über MS-OS/2-fähige Software leider noch nicht geschrieben werden. Aber als aufmerksamer Beobachter der Software-Entwicklungsszenerie sehen wir alle Anzeichen, daß die überwiegende Mehrzahl von Software-Produkten, die gegenwärtig für MS-OS/2 angeboten wird, im nächsten Jahr, das heißt 1989, in MS-OS/2-Versionen verfügbar sein wird. Wie dann diese Software im Multitasking-Betrieb zu handhaben ist, kann den Angaben im Abschnitt zum Programm-Selektor entnommen werden.

14 Zehn Befehle genügen für den Anwender: Wer jetzt mit MS-OS/2 beginnt, tut sich ebenso leicht wie mit MS-DOS

Die Befehle von MS-OS/2 sind für denjenigen, der in MS-DOS bewandert ist, zu einem Großteil bekannt. Von daher wird der Kenner von MS-DOS in der folgenden Darstellung kaum Neues finden. Aber weil es vielleicht auch für ihn wichtig ist zu wissen, was alles gleich geblieben und was neu hinzugekommen ist, kann auch der folgende Abschnitt für ihn von Interesse sein.

Formal wird so vorgegangen: Wir schreiben in einem kurzen Satz die Funktion des Befehls in allgemeiner Art auf. Unter Funktion versteht man die Aufgabe des Befehls - das, was er zu verrichten hat. Dann sagen wir, wo der Befehl vorkommt: im Real-Mode oder Protected-Mode. Je nachdem kann der Befehl nach der Wahl des MS-DOS-Prompt oder des OS/2-Prompt im Programm-Selektor ausgeführt werden. Schließlich notieren wir die sogenannte Syntax der Befehle. Für Syntax können Sie auch Grammatik

sagen - gemeint sind die Regeln, die den Aufbau des Befehlssatzes bestimmen. Bei der Notation der Syntax halten wir uns an die übliche Schreibweise von Handbüchern, die folgendermaßen zu lesen ist:

Zuerst kommt das Befehlswort, dann die Zusätze zum Befehl. Bei den Zusätzen gibt es die Unterscheidung zwischen zwei Arten von Ergänzungen. Die erste Sorte von Ergänzungen bezieht sich auf die Ziel-Objekte des Befehls. Wenn der Befehl zum Beispiel "zeige" heißt, dann könnte "die rote Akte" oder etwas ähnliches ergänzt werden. Solche Zusätze nennt man Optionen oder optionale Ergänzungen (auf Wunsch). In allgemeiner Schreibweise notiert man das so:

Zeige [Objekt]

In der eckigen Klammer steht der Oberbegriff für das, was ergänzt werden kann oder soll. Das Zeigen kann nun auf verschiedene Weise geschehen. Zum Beispiel kann auf einmal gezeigt werden oder häppchenweise oder in einer bestimmten Anordnung. Eine solche Sorte von Ergänzungen gibt die Art und Weise an, wie die Aufgabe zu verrichten ist. Man nennt sie Parameter. Wenn der Buchstabe a für "auf einmal" steht, dann schreibt man das so:

Zeige [Objekt] [/a]

Parameter werden also stets hinter einen Slash (Schrägstrich) geschrieben. Und weil sie optional sind, stehen sie in der allgemeinen Schreibweise ebenfalls in eckigen Klammern. Soweit zur Notation!

Inhaltlich gehen wir folgendermaßen vor: Wir versetzen uns in die Lage dessen, der seinen ersten Personal-Computer mit dem Betriebssystem MS-OS/2 gekauft hat und der noch nichts weiß - weder wie man die Befehle eingibt noch welche Befehle es gibt und der auch nicht weiß, was er an Software auf seinem Computer hat. So wollen wir den Beginner sowohl leiten, seinen Personal-Computer unter MS-OS/2 kennenzulernen, als auch sich die wichtigsten Befehle anzueignen, mit denen er die notwendige Verwaltungsarbeit machen kann. Wir wollen ihn so einführen, daß er zugleich seinen Personal-Computer entdeckt.

Dazu sind noch einige Vorbemerkungen zu den Aufgaben eines Betriebssystems nötig. Wie der Name schon sagt, verwaltet das Betriebssystem einen

Betrieb. Ob dieser Betrieb ein Maschinen-Betrieb oder ein Betrieb unter Menschen ist, wie wir ihn aus dem Alltag kennen, ist gleichgültig. In jedem Falle ist er organisiert. Und diese Organisation ist die Aufgabe des Betriebssystems. Aber es ist auch zugleich seine Aufgabe, diese Organisation transparent werden zu lassen. Das heißt, es geht um den Durchblick: Was ist organisiert und wie ist es dargestellt?

Für den Benutzer ist wichtig, daß er sich in die innere Organisation, die das Betriebssystem leistet, Einblick verschaffen kann. MS-OS/2 bietet dafür einige Befehle, die dies auf den verschiedenen Ebenen leisten. Der dir-Befehl macht dies für die Speicherverwaltung auf der Festplatte oder den Disketten. Mit ihm kann man ein Verzeichnis aller aktuellen Dateien anfordern und erhält dann eine Liste von Datei-Namen. Dateien muß man sich wie Akten oder Dokumente vorstellen, die in bestimmten Abteilungen des Betriebes aufbewahrt sind. Die Abteilungen heißen beim Personal-Computer auch Verzeichnisse, denn Abteilungen - und das gilt auch für die alltäglichen Betriebe - sind mit einem Verzeichnis der Dinge, die da verwaltet werden, beschreibbar. Der englische Ausdruck directory trifft viel mehr den Sachverhalt der Abteilung, als das der Ausdruck Verzeichnis tut. Directories und Subdirectories sind die Direktorien von Abteilung und Unterabteilung.

Verzeichnisse sind also bestimmte Bereiche auf unseren Speichermedien. Diese Bereiche sind teilweise ineinander verschachtelt. Auf der obersten Ebene der Organisation von Speicherbereichen liegt die "Chefetage"; sie heißt die Wurzel oder mit dem englischen Ausdruck die Root bzw. das Root-Verzeichnis. In der Wurzel des Ganzen befindet man sich, wenn man aus dem Programm-Selektor erstmals ein Prompt aufruft.

Funktion: DIR listet die Dateien eines Verzeichnisses auf.

Wo: im Real-Mode und Protected-Mode

Syntax im Real-Mode:

dir [d:] [Pfad] [/p] [/w]

Syntax im Protected-Mode:

dir [d:] [Pfad] [...] [/p] [/w]

Ziel: Sie wollen sehen, was alles im Wurzel-Verzeichnis liegt.

Weg: dir

Resultat: Sie erhalten eine Liste aller Dateien zeilenweise aufgeführt mit der Angabe des Speicherplatzes in Byte und dem Datum und der Uhrzeit der letzten Bearbeitung der jeweiligen Datei.

Bemerkung: Die Eingabe des dir-Kommandos ohne Parameter bewirkt die Anzeige sämtlicher Dateien des aktuellen Verzeichnisses. Aktuell ist jetzt, nachdem man vom Selektor aus auf das MS-DOS Command Prompt oder auf das OS/2 Command Prompt gewechselt ist, die Festplatte.

Sie können die Wildcards (die Wilden - das sind die Jolly Joker beim Kartenspiel, die man für alles einsetzen kann!) * und ? verwenden. Das Fragezeichen kann jeden beliebigen Buchstaben und das Sternchen jede beliebige Buchstabenfolge vertreten. In diesem Sinne ist der folgende Befehl mit dem einfachen dir-Kommando von oben gleichbedeutend.

dir *.*

Bemerkung: Dateinamen bestehen zumeist aus zwei Teilen, die durch einen Punkt voneinander getrennt sind. Der Teil vor dem Punkt kann acht Buchstaben lang sein; der Teil hinter dem Punkt, die sogenannte Erweiterung oder Extension des Namens, kann maximal drei Buchstaben lang sein. Das läßt sich sinnvoll mit Vor- und Nachnamen bei Personen vergleichen. Denn auch hier gibt der Nachname die Familie an, zu der die Datei gehört. So bilden zum Beispiel alle Dateien mit der Erweiterung .exe die Familie der auf der Betriebssystemebene ausführbaren Programme. Der Befehl "dir *.*" bedeutet nun, daß alle Dateien gezeigt werden sollen, gleichgültig welchen Vor- oder Nachnamen sie haben. Denn der Stern vertritt jede Buchstabenfolge (Zeichenkette).

Weil bei diesem Kommando die Liste der Dateien, wenn sie aus mehr als 21 Kandidaten besteht, nicht vollständig auf dem Bildschirm sichtbar ist, empfiehlt sich die Verwendung des Parameters:

/w

Ziel: Sie wollen möglichst viele Dateien auf dem Bildschirm sehen.

Weg: dir /w

Resultat: Es werden nur die Dateinamen ohne die zusätzlichen Daten zu Größe und Zeit angezeigt. Pro Zeile werden fünf Dateinamen ausgegeben. Das sieht so aus: (Bild 14.1)

In der Liste wird der Nachname der Dateien abgetrennt und ohne Punkt dargestellt, damit man einen tabellarischen Überblick hat. Will man gezielter hinschauen, dann kann man sich auch selektiv, das heißt in einer Auswahl, Dateien zeigen lassen.

Ziel: Sie wollen nur die Dateien vom Typ "exe" sehen.

Weg: Geben Sie nur die Kennzeichnung für die Datei-Erweiterung an:

dir .exe oder dir *.exe

Als Resultat erhalten Sie die folgende Liste: (Bild 14.2)

Sie können auch umgekehrt auswählen, indem Sie vom Vornamen ausgehen, Teile davon angeben und den Rest des Vornamens und den Nachnamen mit dem Stern-Symbol variabel halten.

Ziel: Sie wollen alle Dateien sehen, die zum Programm-Selektor gehören. Sie wissen, daß deren Name stets mit "shell" beginnt.

Weg: dir shell*.*

```

[C:\]dir *.exe

HARDERR  EXE      16304  11-01-87  12:00a
SHELL    EXE      36928  11-01-87  12:00a
SWAPPER  EXE       4150   10-24-87   2:27p
CMD       EXE     57648  10-26-87   1:00p
          4 Datei(en) 11923456 Bytes frei
    
```

Bild 14.1


```

C:\>dir *.
SYSTEM      <DIR>      1-12-88   1:20p
HG          <DIR>      1-12-88   4:16p
PBRUSH      <DIR>      1-13-88   1:25p
NORTON      <DIR>      1-13-88   2:38p
WS          <DIR>      1-14-88  10:07a
TB          <DIR>      1-14-88  10:54a
NC          <DIR>      1-18-88   5:12p
TEMP        <DIR>      2-29-88   1:21p
RHH         <DIR>      3-09-88   4:11p
DS2         <DIR>      3-11-88   2:26p
WORD        <DIR>      3-17-88   2:48p
RBSYS       <DIR>      3-17-88   2:48p
CHART       <DIR>      3-17-88   2:48p
MP          <DIR>      3-17-88   2:49p
HAMB        <DIR>      4-26-88   1:30p
CHIP        <DIR>      5-13-88   8:45a
VIDERE      <DIR>      5-13-88  10:06a
SEM         <DIR>      5-13-88   2:30p
HAMB2       <DIR>      5-21-88   9:26p
19 Datei(en) 11923456 Bytes frei

```

Bild 14.2

Als Resultat erhalten Sie die folgende Liste: (Bild 14.3)

Ziel: Wir haben in Bild 14.1 eine Reihe von Namen entdeckt, die keine Erweiterung hatten. Das wollen wir uns genauer ansehen.

Weg: dir *.

Wir lassen also einfach den Nachnamen hinter dem Punkt weg.

Resultat: Wir sehen eine Liste mit Namen, die alle anstelle der Erweiterung die Kennzeichnung <DIR> haben. Das sind keine echten Dateien, sondern Abteilungen bzw. die oben schon erwähnten Verzeichnisse. Wir werden sehen, daß in ihnen weitere Dateien liegen (Bild 14.4).

Ziel: Sie wollen in das Verzeichnis von Word schauen und eine Liste aller Briefe sehen.

Weg: dir c:\word\texte\Brief*.*

Wichtige Bemerkung: Das ist schon eine komplexere Auswahl von Dateien, die gelistet werden sollen. Sie wählen erst die Festplatte (c:), dann das Verzeichnis word und schließlich das Unterverzeichnis Texte, aus dem Sie nur die Dateien gezeigt haben wollen, die mit dem Namensteil Brief beginnen. Das Laufwerk wird stets mit seiner Buchstabenbezeichnung, gefolgt von einem Doppelpunkt, angegeben. In der oben angeführten allgemeinen Schreibweise ist das durch [d:] kenntlich gemacht. Das d verwendet man in diesem Zusammenhang, weil es die Assoziation von Diskette nahelegt: a: steht nämlich für das erste Diskettenlaufwerk und b: für das zweite Diskettenlaufwerk, c: schließlich für die Festplatte usw. Verzeichnisse und Unterverzeichnisse erhalten als Erkennungszeichen

```

C:\>dir shell.*.*
SHELL      EXE      36928  11-01-87  12:00a
SHELL      LIB      17348  11-01-87  12:00a
SHELL11F   AIF        269   11-01-87  12:00a
SHELL11F   AII        86   11-01-87  12:00a
4 Datei(en) 11923456 Bytes frei

```

Bild 14.3


```

C:\>dir /w

volume in drive C has no label.
Directory of C:\


COMMAND  COM  CONFIG  OS2  SYSTEM  AUTOEXEC  ZOS  HG
PBRUSH   COM  NORTON  WS  SYSTEM  TB  NC
TEMP     COM  RRH     WS  CLOCK01  SYS  DISK01  SYS  KBD01  SYS
PRINT01  SYS  SCREEN01  SYS  COUNTRY  SYS  HARDERR  EXE  SHELL  EXE
SHELL    LIB  SHELL11F  AIF  SHELL11F  AII  SWAPPER  EXE  CMD  EXE
OS0001   MSG  ZDS      MSG  OS2      ZDSH  MSG  OS0001H  MSG
FORMATS  TBL  CONFIG  BAK  STARTUP  BAK  INITENU  BAK  AUTOEXEC  BAK
SWAPPER  DAT  CONFIG  SYS  STARTUP  CMD  INITENU  CMD  AUTOEXEC  SIK
WORD     COM  RBSYS   CHART  MP  AUTOEXEC  BAT
HAMB     COM  CHIP    VIDERE  SEM  AUTOEXEC  BAT
CONFIG   SYS  OS2     TXT  SEM  HAMB2

52 File(s) 11921408 bytes free

C:\>

```

Bild 14.4

an der ersten Stelle den sogenannten Backslash, das ist der umgekehrte Schrägstrich, der auf den meisten Tastaturen erzeugt werden kann, indem man die Alt-Strg-Tastenkombination gedrückt hält und eine weitere Taste anschlägt. Diese ist an der Seite dann mit dem Backslash-Zeichen markiert. In jedem Falle aber kann der Backslash so geschrieben werden, daß man die Alt-Taste niedergedrückt hält und auf dem Zahlenblock der Tastatur 92 eingibt. Läßt man die Alt-Taste dann los, so erscheint der Backslash auf dem Bildschirm.

Man nennt die Kette von Verzeichnissen, in die man einschauen will, auch den Pfad. In unserem Beispiel hier ist der Pfad \word\texte. Damit ist auch geklärt, was die allgemeine Notation [Pfad] oben bedeutet. Bleibt noch die Schreibweise [...] zu erklären: Sie bezieht sich auf die vorangehende Angabe und weist darauf hin, daß beliebig viele Angaben des gleichen Types folgen dürfen.

NEU in MS-OS/2: Sie können im Protected-Mode jetzt mehrere Pfade angeben und erhalten so gemischte Listen. Zum Beispiel so:

```

C:\OS2>dir

.           <DIR>      3-11-88  2:26p
..          <DIR>      3-11-88  2:26p
LIB         <DIR>      3-11-88  2:26p
DEV         <DIR>      3-11-88  2:26p
BIN         <DIR>      3-11-88  2:27p
PBIN        <DIR>      3-11-88  2:27p
RBIN        <DIR>      3-11-88  2:27p
SPOOL       <DIR>      3-11-88  2:30p

8 Datei(en) 11923456 Bytes frei

```

Bild 14.5


```
dir c:\word\texte c:\rbsys\daten
```

Wir kommen nun zur näheren Erklärung von Verzeichnissen. Wie überall ist es hier auch so: Man hat verstanden, worum es geht, wenn man damit umgehen kann. Der folgende Befehl läßt uns durch die Abteilungen unserer Speicherverwaltung "spazieren".

Funktion: CHDIR (CD) ändert das aktuelle Verzeichnis bzw. zeigt das aktuelle Verzeichnis an.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

```
chdir [d:] Pfad oder cd [d:] Pfad
```

Ziel: Sie wollen das Verzeichnis wechseln (change directory) und in die Abteilung OS2 gehen und sich dort umsehen.

Weg: cd os2

Geben Sie anschließend das Kommando dir ein.

Resultat: Sie erhalten die folgende Liste, die wiederum nur weitere Verzeichnisse enthält (Bild 14.5).

Bemerkungen: Wenn wie hier schon von der Installation her eine Baumstruktur bzw. Hierarchie von Verzeichnissen und Unterverzeichnissen auf Ihrem Datenträger angelegt ist, dann legt MS-OS/2 zwei spezielle Dateien an, die mit "." und ".." gekennzeichnet sind und für die Organisation gebraucht werden.

Die weiteren Verzeichnisse stammen aus der Standard-Installation von MS-OS/2, wie sie von Zenith angepaßt und eingerichtet ist. In Kürze sei hier erläutert, wie alles im einzelnen angelegt ist.

Das Verzeichnis LIB enthält die Dateien, die für das Dynamic-Link-Verfahren, das MS-OS/2 für den Ablauf von Programmen verwendet, notwendig sind. Etwas verkürzt ausgedrückt, aber ausreichend für eine Grundvorstellung von diesem Verfahren heißt das, daß MS-OS/2 die Programme erst beim Starten endgültig lauffähig macht.

Das Verzeichnis DEV enthält alle Device-Treiber (Geräte-Treiber), das sind die Schnittstellen zum Beispiel zum Drucker, zur Maus oder zur Tastatur.

Das Verzeichnis BIN enthält die externen Befehle, die dem Real-Mode und dem Protected-Mode gemeinsam sind. Externe Befehle sind solche, die als eigenständige Dateien vorliegen und nicht schon im Betriebssystemskern enthalten sind.

Das Verzeichnis RBIN enthält die externen Befehle nur für den Real-Mode.

Das Verzeichnis PBIN enthält die externen Befehle, die nur für den Protected-Mode vorgesehen sind.

Das Verzeichnis SPOOL ist normalerweise leer und nur dafür eingerichtet, Dateien aufzunehmen, die im Multitasking-Betrieb für den Druck vorgesehen, aber noch nicht an der Reihe sind.

```
[C:\OS2\PBIN] dir/p
```

.	<DIR>	3-11-88	2:27p
..	<DIR>	3-11-88	2:27p
FDISK	COM	37576	12-04-87 12:06p
ANSI	EXE	9984	10-26-87 4:01p
CONFIGUR	EXE	27226	12-11-87 11:58a
KEYB	COM	12636	10-26-87 4:08p
SHIP	EXE	10190	11-25-87 5:47p
SPOOL	EXE	77370	10-26-87 4:14p
HELP	CMD	414	12-12-87 11:10a
10 Datei(en)		11921408	Bytes frei

Bild 14.6

Ziel: Sie wollen nun noch eine Ebene tiefer ins Verzeichnis PBIN gehen.

Weg: cd pbin

Geben Sie anschließend das Kommando dir /p ein. Wenn Sie den Parameter /p im Befehl mitverwenden, dann hat das die folgende Bedeutung: Es wird jeweils nur eine Bildschirmseite ausgegeben. Indem Sie eine beliebige Taste betätigen, können Sie weiterblättern. Die Bildschirm-Ausgabe wird dann fortgesetzt. Das hat immer dann Sinn, wenn Sie nicht genau wissen, wie lang die Liste der Dateien ist, und wenn Sie nicht auf die zusätzlichen Angaben verzichten wollen, wie das bei dem Parameter /w der Fall ist.

Resultat: Sie erhalten die folgende Liste, die Protected-Mode Befehle enthält (Bild 14.6).

Bemerkung: Immer tiefer in die verschachtelten Abteilungen zu gehen ist kein Problem. Das macht stets der einfache cd-Befehl. Aber man muß ja auch wieder zurückgehen können. Dazu die folgenden Regeln:

Das Kommando "cd .." macht das derzeitige "Elternverzeichnis", das übergeordnete directory, zum aktuellen Verzeichnis. Das Kommando "cd \" springt ins Wurzelverzeichnis und macht es zum aktuellen Ver-

zeichnis. Das Wurzelverzeichnis ist im übrigen auch das Verzeichnis, das nach dem Start von MS-OS/2 als erstes angezeigt wird. Bei der Eingabe von "cd" ohne einen Pfad wird das aktuelle Verzeichnis angezeigt, in dem Sie sich gerade befinden. Während "cd .." und "cd \" gleichsam nach oben springen in Richtung Chef-Etage, geht der Befehl "cd \Pfad" seitwärts und man landet in einem Verzeichnis auf derselben Hierarchie-Ebene. Beispiel:

Ziel: Sie wollen in das auf derselben Ebene bzw. im selben übergeordneten Verzeichnis liegende Unterverzeichnis DEV gehen und sich die Treiber ansehen. Das ist auf zwei Arten möglich:

Weg 1: cd .. - dann sind Sie im Verzeichnis OS2

cd dev - dann sind Sie im Verzeichnis DEV

Weg 2: cd \os2\dev

Bemerkung: Beim Weg 2 muß man den gesamten Pfad von der Wurzel aus angeben.

Geben Sie anschließend das Kommando dir ein.

Resultat: Sie erhalten die folgende Liste, die Protected-Mode Befehle enthält (Bild 14.7).

```

[C:\OS2\DEV]dir
.                <DIR>          3-11-88   2:26p
..               <DIR>          3-11-88   2:26p
ANSI             SYS          1694  11-01-87  12:00a
COM01            SYS          9268  12-09-87  11:29a
EGA              SYS          2110  11-01-87  12:00a
UDISK            SYS          4662  11-01-87  12:00a
EXTDSKDD         SYS          1860  12-09-87   4:10p
POINTDD          SYS          5886  11-01-87  12:00a
KEYBOARD         DCP          85917 11-01-87  12:00a
UIOTBL           DCP          52162 11-01-87  12:00a
MOUSEA00         SYS          16950 12-03-87   3:09p
MOUSEA01         SYS          16438 12-03-87   3:09p
MOUSEA02         SYS          16438 11-01-87  12:00a
MOUSEA03         SYS          16438 11-01-87  12:00a
MOUSEA04         SYS          16438 11-01-87  12:00a
4201             DCP          17081 11-01-87  12:00a
5202             DCP           416  11-01-87  12:00a
17 Datei(en) 11921408 Bytes frei
  
```

Bild 14.7

Funktion: TREE zeigt den Pfad und optional (das heißt bei Wahl) die Inhalte aller Verzeichnisse und Unterverzeichnisse eines Laufwerks an.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

TREE [d:] [/f]

Bemerkung: Bei so komplexen Strukturen und Bewegungsmöglichkeiten ist es unerlässlich, einen Befehl zur Verfügung zu haben, der es erlaubt, die gesamte Struktur im Aufbau der Speicher-Verwaltung anzuzeigen und für den Benutzer durchsichtig zu machen. Das tree-Kommando gibt den vollständigen Pfad jedes Verzeichnisses zusammen mit den Namen seiner Unterverzeichnisse an.

Parameter: Der Parameter /f bewirkt, daß zusätzlich die Dateien des jeweiligen Verzeichnisses angezeigt werden.

Eine andere Möglichkeit, die Unterverzeichnisse des aktuellen Verzeichnisses auf dem Bildschirm auszugeben, stellt die Eingabe von "dir *." dar. Allerdings werden dabei auch alle Dateien angezeigt, die keine Erweiterungen besitzen, und Sie können den Strukturverlauf als Baum nicht einsehen. Verzeichnisse sind in der Liste der Dateien durch <DIR> gekennzeichnet.

Beispiel: Wird die Anzeige aller Verzeichnisse und Unterverzeichnisse auf dem Bildschirm gewünscht, genügt die Eingabe von:

tree

Wenn Sie zusätzlich die Dateien aufgelistet haben wollen, die sich in den jeweiligen Verzeichnissen auf dem Laufwerk C: befinden, so müssen Sie die folgende Form des tree-Kommandos wählen:

tree c:/f | more

Dabei sorgt das anschließende more-Kommando dafür, daß jeweils nur eine volle Bildschirmseite ausgegeben wird. Das sieht dann so aus wie bei der Verwendung des Parameters /p im dir-Kommando. Die vollständige Erklärung lautet folgendermaßen:

Funktion: MORE ermöglicht eine seitenweise Ausgabe von größeren Datenmengen auf dem Bildschirm.

Wo: im Real-Mode und im Protected-Mode

Syntax im Real-Mode und Protected-Mode:

more < Quelle oder

Quelle | more

Bemerkung: "Quelle" bezeichnet ein Kommando wie zum Beispiel den oben genannten tree-Befehl. Wir werden später sehen, daß der type-Befehl am besten auch zusammen mit dem more-Befehl verwendet wird. Das Zeichen < heißt Umleitung, denn es leitet die Ausgabe von tree zum Beispiel an den more-Befehl weiter. Das Zeichen | ist der Pipe-Operator, der zwei Befehle derart verkettet, daß der Output des ersten zum Input des nächsten wird.

Das more-Kommando nennt man einen Filter. Er liest von der sogenannten Standardeingabe (von einer Pipe oder einer umgeleiteten Datei), "filtert" die Eingabe und gibt jeweils eine komplette Bildschirmseite an Daten aus. Das more-Kommando ist vor allem zum Lesen längerer Dateien oder Befehlsausgaben - wie beim tree-Befehl - geeignet.

Als Quelle kann zum Beispiel auch das dir-Kommando verwendet werden. Nach der Ausgabe einer Bildschirmseite meldet sich das more-Kommando auf der untersten Bildschirmzeile. Wenn Sie dann die Return-Taste betätigt haben, wird jeweils die Ausgabe der nächsten Seite der Datei veranlaßt. Bei der Eingabe der Tastenkombination Strg-C (Ctrl-C) wird das more-Kommando abgebrochen.

Das more-Kommando legt für die auszugebenden Daten eine temporäre Datei auf der Diskette an. Falls die Diskette voll oder schreibgeschützt ist, funktioniert das more-Kommando nicht.

Wir wissen nun, wie man sich in den Abteilungen der Speicher-Verwaltung des Betriebssystems bewegt (cd), wie man die Organisationsstruktur dieser Verwaltung einsehen kann (tree) und wie man sich zeigen läßt, welche Akten in den Abteilungen liegen (dir). Das englische Wort file für Datei heißt übrigens ursprünglich nichts anderes als Aktenbündel.

Jetzt wollen wir sehen, wie man solche Abteilungen anlegt.

Funktion: MKDIR (MD) erstellt ein neues Unterverzeichnis im aktuellen Verzeichnis.

Wo: im Real-Mode und Protected-Mode

Syntax im Real-Mode:

`mkdir [d:] Pfad`

Syntax im Protected-Mode:

`mkdir [d:] Pfad [...]`

Ziel: Sie wollen im Wurzel-Verzeichnis, in dem Sie sich aktuell aufhalten, die Unterverzeichnisse Übung und Test anlegen. Dazu geben Sie den folgenden Befehl ein:

Weg im Real-Mode: `md uebung`

und `md Test`

Weg im Protected-Mode: `md uebung Test`

Neu im Protected-Mode: es können mit einem `mkdir`-Kommando jetzt mehrere Verzeichnisse erzeugt werden, indem mehrere Pfade angegeben werden.

Bemerkung: Das `mkdir`-Kommando ermöglicht den Aufbau einer baumartigen Verzeichnisstruktur über mehrere Ebenen. Verzeichnisse, die mit dem `mkdir`-Kommando erstellt werden, sind immer Unterverzeichnisse des aktuellen Verzeichnisses, falls nicht ausdrücklich ein anderer Pfad angegeben wird. Bei der Befehlseingabe genügt die Schreibweise `md`, die oben in Klammern angegeben ist.

Nun haben wir zwei neue Verzeichnisse, in denen wir auch etwas ablegen wollen.

Funktion: COPY kopiert eine oder mehrere Dateien auf ein anderes Laufwerk oder in ein anderes Verzeichnis. Außerdem können Dateien verkettet und auf demselben Laufwerk oder in demselben Verzeichnis kopiert werden. Dabei ist die Angabe von zwei Dateinamen erforderlich.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

`copy [d:] [Pfad] Dateiname [d:] [Pfad] [/v] [/a] [/b]`

kopiert eine Datei aus einem Verzeichnis auf Laufwerk `d:` in ein Verzeichnis eines anderen oder desselben Laufwerks.

`copy [d:] [Pfad] Dateiname + [d:] [Pfad] Dateiname`

verkettet zwei Dateien. Die Verkettung von mehreren Dateien ist auch möglich.

Bemerkung: Der `copy`-Befehl ist für das Übertragen von Daten zuständig. Man muß zuerst die Dateien kopieren und kann sie dann später, wenn man dies so will, an der ursprünglichen Stelle löschen. Der erste Dateiname bezeichnet die Quellen- oder Ursprungsdatei, der zweite die Zieldatei.

Ziel: Sie wollen alle Dateien von Ihrer Betriebssystem-Diskette vom Laufwerk `A:` in das Verzeichnis `TEST` auf der Festplatte übertragen.

Weg 1: `copy a:*. * c:\test`

Weg 2: `cd \test`

und `copy a:*. *`

Bemerkung: Wenn Sie keine Zieldatei spezifizieren, dann wird in das aktuelle Verzeichnis auf dem Standardlaufwerk kopiert. Das geht allerdings nur, wenn das aktuelle Verzeichnis von dem Verzeichnis, in dem die Quellendatei liegt, verschieden ist. Sonst müßte die Quellendatei auf sich selbst kopiert werden, was nicht möglich ist. Dementsprechend erscheint eine Fehlermeldung. Sind also beim `copy`-Befehl die Verzeichnisse unterschieden, dann werden Name und Erstellungsdatum und -zeit von der Originaldatei übernommen.

Man kann mit dem `copy`-Befehl einige sinnvolle Parameter verwenden. Der Parameter `/v` bewirkt, daß die korrekte Übertragung der Daten geprüft wird. Das entspricht einem Datenvergleich zwischen Ursprungs- und Zieldatei, wie er auch durch den `comp`-Befehl (siehe an späterer Stelle) durchgeführt wird.

Die beiden Parameter `/a` und `/b` bedeuten, daß entweder ASCII- oder Binärdateien kopiert werden. Das hat die folgende Auswirkung bei der Quellendatei:

Bei dem Parameter /a wird die Datei als ASCII-Datei (Text-Datei) behandelt. Die Daten werden bis zur ersten Datei-Endemarke (End of file/EOF-Zeichen), die in der Quellendatei gefunden wird, kopiert. Die Datei-Endemarke ist bei der Kopie nicht eingeschlossen. Beim Parameter /b wird die gesamte Datei kopiert.

Die Auswirkung bei der Zieldatei sieht mit dem Parameter /a folgendermaßen aus: Eine Datei-Endemarke wird an das letzte Zeichen der Datei angefügt. Beim Parameter /b wird keine Dateiendemarke angefügt.

Spezielle Anmerkung, die für den Normalgebrauch nicht so wichtig ist: Wenn Sie den Parameter /b verwenden, dann werden die ASCII-Zeichen unter 20H (hexadezimal) nicht als Steuerzeichen interpretiert. Wenn Sie zum Beispiel ausführbare Programme (in diesen Dateien treten im allgemeinen Bytes unter 20H auf) kopieren wollen, dann müssen Sie diesen Parameter verwenden, um zu vermeiden, daß die Datei fehlerhaft oder unvollständig kopiert wird. Beim ersten Auftreten eines EOF-Zeichens (1A hexadezimal) würde nämlich dann der Kopiervorgang abgebrochen. Mit dem Parameter /b wird dagegen der Kopiervorgang erst dann beendet, wenn die Anzahl von Zeichen gesendet wurde, die als Dateigröße im Verzeichnis angegeben ist.

Sollen Dateien verkettet werden, so wird der Parameter /a angenommen, falls keine anderen Angaben gemacht wurden.

Ziel: Sie wollen zwei Autoexec-Dateien verkettet und danach in einem Textverarbeitungsprogramm zu einer neuen verarbeiten.

Weg: Copy autoexec.bat + autoexec.os2 autoexec.txt

Bemerkung: Wenn Sie bei einer Verkettung von Dateien keine Zieldatei angeben, so ist die zuerst angegebene Datei gleichzeitig die Zieldatei. In den meisten Fällen jedoch empfiehlt es sich, die Zieldatei unabhängig von den Ursprungsdateien neu zu erstellen.

Die alltäglichste Funktion des copy-Befehles besteht darin, Dateien auf der Festplatte in Sicherheitskopien auf einer Diskette aufzubewahren. Dazu müssen Disketten nicht nur gekauft, sondern auch zubereitet werden. Diese Zubereitung nennt man Formatieren.

Funktion: FORMAT formatiert die Diskette oder Festplatte im angegebenen Laufwerk.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

format [d:] [/4] [/s] [/t:Spuren] [/n:Sektoren] [/v:Name]

Bemerkung: Das format-Kommando legt die Einteilung der Diskette in Spuren und Sektoren fest. Das ist so, als ob man ein Buch mit leeren, aber durchnummerierten Seiten herstellt. Format erzeugt dann ein Verzeichnis und die sogenannte Dateizuordnungstabelle auf der Diskette oder Festplatte. Die Zuordnungstabelle ist so etwas ähnliches wie das Inhaltsverzeichnis oder das Schlagwortregister in einem Buch. Jede neue Diskette muß formatiert werden, bevor sie vom System benutzt werden kann.

Warnung: Beim Formatieren werden alle Daten, die sich gegebenenfalls auf der Diskette befinden, zerstört. Vermeiden Sie deshalb unbedingt, in den Befehl "format c:" zu geraten. Sollte Ihnen dies dennoch einmal widerfahren und Sie stehen vor der letzten Sicherheitsabfrage, ob Sie weitermachen wollen oder nicht, dann schalten Sie lieber Ihren Rechner aus, bevor Sie irgendeine falsche Taste betätigen. Beim Ausschalten des Gerätes gehen jedenfalls keine gespeicherten Programme oder Daten verloren. Außerdem ist noch wichtig zu wissen, daß die Laufwerkszuweisungen, die durch den assign-Befehl gesetzt sind, vom format-Kommando ignoriert werden.

Ziel: Wir wollen jetzt eine Diskette in Laufwerk B: formatieren und gleichzeitig die Systemdateien auf die neu formatierte Diskette übertragen. Dazu genügt die Eingabe von:

Weg: format b:/s

Bemerkung: Mit dem Parameter /s kopiert man gleichzeitig die Betriebssystemdateien OS2DOS.COM und OS2BIO.COM sowie alle anderen Dateien, die in der Datei formats.tbl aufgeführt sind, von der Diskette im Standardlaufwerk bzw. von der Festplatte auf die neu formatierte Diskette oder Festplatte. Damit wird die Diskette oder Festplatte bootfähig, das heißt Ihr Rechner kann mit ihr hochgefahren werden, was mit einer normalen Diskette nicht geht.

Auch der Parameter /4 besitzt eine wichtige Anwendung.

Ziel: Sie wollen eine 5-1/4-Zoll-Diskette in einem High-Capacity-Laufwerk auf 360 Kilobyte Speicherplatz formatieren, um mit einem normalen MS-DOS-Personal-Computer Daten tauschen zu können. Denn dieser kann in der Regel nur 360-Kilobyte-Disketten lesen, während Ihr MS-OS/2-Personal-Computer möglicherweise nur ein 1,2 Megabyte Laufwerk hat.

Weg: format a:/4

Bemerkung: Ohne diesen Parameter wird "high density", das heißt für erhöhte Datendichte, formatiert. Man erhält dann 1,2 Megabyte Speicherplatz. Sie sollten dann aber Disketten von besonderer Qualität verwenden.

Ziel: Sie haben vielleicht ein 3-1/2-Zoll-Laufwerk und wollen eine entsprechende Diskette formatieren.

Weg: format a: /t:80 /n:18

Bemerkung: Für 720-Kilobyte- und 1,44-Megabyte-Disketten ist der Wert des Parameters [t:Spuren] jeweils 80 (/t:80). Spuren bezeichnet die Anzahl der "Rillen" - um eine Analogie zur Schallplatte herzustellen - auf der Diskette. Mit diesem Parameter werden 3-1/2-Zoll-Disketten entsprechend der angegebenen Anzahl der Spuren formatiert. Aber man benötigt darüber hinaus den Parameter [/n:Sektoren]. Sektoren bezeichnet die Anzahl der Kreissektoren, das sind gleiche Abschnitte auf einem Kreis, pro Spur. Für 720 Kilobyte-Disketten ist dieser Wert 9 (/n:9) und für 1,44 Megabyte-Disketten - wie in unserem Beispiel - ist dieser Wert 18 (/n:18).

Ziel: Sie wollen der Diskette, die Sie formatieren, gleichzeitig einen Namen geben.

Weg: format a: /v:Karl

Bemerkung: Karl ist der Diskettenname, der dem Benutzer zur Kennzeichnung der neu formatierten Diskette dient. Dieser Name kann bis zu 11 Zeichen lang sein.

Weitere Bemerkung: Sie können natürlich auch mehrere dieser Parameter auf einmal verwenden. Wenn der äußerst seltene Fall eintritt, daß eine bereits formatierte Festplatte neu formatiert werden

soll, wird der Benutzer zunächst nach dem Namen gefragt, den er der Festplatte gegeben hat. Wenn die Platte keinen Namen hat, braucht nur die Eingabetaste betätigt werden. Nach der korrekten Eingabe des Namens fragt das System an, ob die Neuformatierung der Platte tatsächlich gewünscht wird. Nachdem der Formatierungsvorgang beendet ist, wird der Zustand für die neu formatierte Diskette oder Festplatte gemeldet.

Mit der nun formatierten Diskette können Sie verfahren, wie mit der Festplatte. Sie haben da nur weniger Speicherplatz.

Ziel: Sie können zum Beispiel ein Verzeichnis DEV für die Device-Treiber von MS-OS/2 anlegen.

Weg: md A:\dev & copy c:\os2\dev a:\dev /v

NEU in MS-OS/2: Das &-Zeichen ist ein Trennoperator für die Befehlseingabe im Protected-Mode. Sie können also damit mehr als einen Befehl - von einander wohlgetrennt - in eine Zeile schreiben. Das ist sehr sinnvoll für ein Multitasking-Betriebssystem, denn auf diese Weise kann man längere Prozesse in den Hintergrund verlegen und im Vordergrund am Bildschirm andere Arbeit tun.

Funktion: REN (RENAME) ändert den Namen einer Datei.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

ren [d:] [Pfad] Dateiname.alt Dateiname.neu

Ziel: Sie wollen im Unterverzeichnis test, das wir vorhin angelegt haben, die Datei autoexec.txt in autoexec.sik (sik für Sicherheitskopie) umbenennen. Dazu können Sie den folgenden Befehl verwenden:

Weg: REN test\autoexec.txt autoexec.sik

oder REN test\autoexec.txt *.sik

Bemerkung: Das rename-Kommando benennt eine Datei um, die innerhalb eines Verzeichnisses von mehreren Dateien liegt. Bei der Angabe des Dateinamens können Wildcards (* und ?) verwendet werden. Damit kann man mehrere Dateien umbenennen. Das rename-Kommando ändert dann die Namen aller

Dateien, die mit dem ersten Dateinamen im Kern, das ist die Bezeichnung jenseits der Wildcards, übereinstimmen. Wenn Sie auch im zweiten Dateinamen Wildcards verwenden, so bleiben die damit gekennzeichneten Namensteile unverändert.

Das ren-Kommando kann natürlich nicht dazu benutzt werden, Dateien von einem Laufwerk oder Verzeichnis in ein anderes zu übertragen. Dazu haben wir ja den copy-Befehl. Wenn man trotzdem versucht, vor dem zweiten Dateinamen ein Laufwerk oder einen Pfad anzugeben, so erfolgt eine Fehlermeldung und die Umbenennung unterbleibt.

Eine wichtige zweite Eingriffsmöglichkeit in die generelle Dateiverwaltung des Betriebssystems ist das Löschen von Dateien. Wie die anderen bisher genannten Befehle sich gegenüber MS-DOS nicht verändert haben, so auch der del-Befehl.

Funktion: DEL löscht die angegebenen Dateien.

Wo: im Real-Mode und Protected-Mode

Syntax im Real-Mode:

del [d:] [Pfad] Dateiname

Syntax im Protected-Mode:

del [d:] [Pfad] Dateiname [...]

NEU im Protected-Mode: Sie können nun eine ganze Liste von Dateien in einem Befehl löschen.

Ziel: Sie wollen zum Beispiel im Protected-Mode alle Dateien mit der Erweiterung .asm im aktuellen Verzeichnis und die Datei prod.txt auf Laufwerk B: löschen.

Weg: del *.asm b: prod.txt

Für den del-Befehl gibt es also auch diese kleine Variante im Protected-Mode, die schon beim mkdir-Befehl genannt wurde: Es können nun mehrere Dateien oder mit Hilfe der Wildcards Datei-Familien aufgelistet werden. MS-OS/2 löscht sie dann nacheinander. Als Synonym zu del können Sie nach wie vor den erase-Befehl verwenden: Eine Reminiszenz an CP/M.

Es ist klar, daß bei der Angabe der Dateinamen die Wildcards * und ? verwendet werden können. Wenn Sie del *.* eingeben, fragt das System wie bisher zurück, ob das Löschen aller Dateien tatsächlich erwünscht ist. Und wenn Sie statt der vollen Adresse - Laufwerk, Pfad und Dateiname - nur den Pfad angeben, nimmt das MS-OS/2 *.* als Dateiname an und löscht nach vorhergehender Rückfrage alle Dateien im angegebenen Verzeichnis. Im Real-Mode gilt immer noch der alte del-Befehl.

Ziel: Sie wollen im aktuellen Verzeichnis alle Dateien löschen, die die Erweiterung .bak (gängig für backup-Dateien) besitzen, dann geben Sie den folgenden Befehl ein:

Weg: del *.bak

Ein weiterer Lösch-Befehl betrifft die Abteilungen in der Speicher-Verwaltung selbst. Was man anlegen kann, muß man natürlich auch wieder beseitigen können. Mit mkdir werden Verzeichnisse, Unterverzeichnisse oder allgemein ausgedrückt: Abteilungen "gemacht", mit rmdir kann man sich im verzweigten Netzwerk dieser Verzeichnisse bewegen, mit tree verschafft man sich den Überblick über diese verzweigte Organisationsstruktur, und schließlich braucht man noch einen Befehl, um Verzeichnisse entfernen zu können, das heißt letztlich um die Organisation von Programmen und Daten ändern zu können. Dazu ist der Befehl "mach das Verzeichnis weg!" notwendig: remove directory!

Funktion: RMDIR (RD) entfernt ein Verzeichnis aus einer beliebigen Ebene in der Organisation aller Verzeichnisse.

Wo: im Real-Mode und Protected-Mode

Syntax im Real-Mode:

rmdir [d:] Pfad

Syntax im Protected-Mode:

rmdir [d:] Pfad [...]

Ziel: Sie haben oben das Verzeichnis uebung angelegt, das wir nun doch nicht gebraucht haben und auch im weiteren nicht mehr brauchen. Wir löschen es. Geben Sie dazu einen der folgenden Befehle ein:

Weg: RD c:\uebung

oder wenn Sie sich in der Wurzel von Laufwerk C befinden, dann reicht:

RD uebung

Bemerkung: Das rmdir-Kommando entfernt ein Verzeichnis, das bis auf die seltsamen Einträge "." und ".." völlig leer sein muß. Diese Einträge beziehen sich auf das Verzeichnis selbst und das übergeordnete Elternverzeichnis, das den Rahmen abgibt. Bevor Sie ein Verzeichnis löschen bzw. entfernen können, müssen Sie also alle Dateien und Unterverzeichnisse, die sich darin befinden, löschen.

NEU im Protected-Mode: Bei dem neuen Befehlsinterpreter hat man auch hier den zusätzlichen Komfort eingebaut: Sie können mehrere Verzeichnisse angeben, die dann nach und nach gelöscht werden. Im Real-Mode, der ja zu der Umgebung von MS-DOS 3.x kompatibel bleiben soll, kann jedoch nur ein Verzeichnis mit einem Kommando gelöscht werden.

Mit dem rd-Befehl sind alle Kommandos zur Handhabung von Verzeichnissen genannt: anlegen (md), wechseln/gehen (cd), Struktur einsehen (tree) und löschen. Außerdem können wir den Inhalt von Verzeichnissen mit dem dir-Befehl sichtbar machen. Was wir noch nicht können, ist dies: in die Dateien hineinsehen. Das erlaubt der type-Befehl.

Funktion: TYPE listet den Inhalt einer Textdatei auf dem Bildschirm auf.

Wo: im Real-Mode und Protected-Mode

Syntax im Real-Mode:

type [d:] [Pfad] Dateiname

Syntax im Protected-Mode:

type [d:] [Pfad] Dateiname [...]

Ziel: Sie wollen die Dateien autoexec.os2 und startup.cmd anzeigen lassen, um sie mit dem, was wir im 7. Abschnitt beschrieben, zu vergleichen. Im Protected-Mode genügt die Eingabe einer Kommandozeile.

Weg: type autoexec.os2 startup.cmd

Im Real-Mode wären dazu zwei type-Kommandos erforderlich.

Weg: type autoexec.os2

und type startup.cmd

NEU im Protected-Mode: Sie können sich eine Vielzahl von Dateien hintereinander zeigen lassen und durchsehen.

Ziel: Die Neubildung des Befehlsinterpreters für Multitasking hat diese feine Erweiterung in die Befehlssprache eingebracht. Die erweiterte Befehlssyntax von cmd.exe macht sich auch schon in den einfachen Befehlen vorteilhaft bemerkbar. Sie können nämlich jetzt auf einem OS/2 Command Prompt immer in den Zwischenzeiten, während auf anderen Prompts für Sie nichts zu tun ist, die Dateien weiter durchsehen. Dazu geben sie dann den more-Befehl zusätzlich ein, damit Sie auch Bildschirmseite für Bildschirmseite blättern können.

Weg: type *.txt | more

Bemerkung: Während dir nur die Dateinamen zeigt, kann man mit dem type-Befehl den Inhalt einer Datei anzeigen lassen. Das type-Kommando zeigt den Inhalt einer Textdatei an, ohne daß die Datei dabei verändert, das heißt editiert, werden kann.

NEU im Protected-Mode: Sie können nicht nur mehrere Dateinamen angeben, die dann in der durch die Kommandozeile festgelegten Reihenfolge ausgegeben werden, sondern dürfen auch Wildcards in den Dateinamen verwenden.

Bemerkungen: Wenn Sie sich eine Datei zeigen lassen wollen, die Tabulatoren enthält, so wird die momentan gültige Einstellung der Tabulatoren (im allgemeinen 8 Leerzeichen) verwendet.

Wenn Sie versuchen, eine Binärdatei oder eine von einer Applikation erstellte Datei am Bildschirm auszugeben, können ungewöhnliche Zeichen - im allgemeinen irgendwelche grafischen Zeichen - auf dem Bildschirm erscheinen. Richtig lesen kann man nur Dateien, die im ASCII-Code gespeichert sind.

Sie haben in diesem 14. Abschnitt die Befehle kennengelernt, ohne die man an dem Personal-Computer ganz einfach nicht arbeiten kann. Andererseits sind es auch die Befehle, die über weite Strecken ausreichen. Und es gibt viele Benutzer, die zumeist oder ausschließlich in der Standard-Software arbeiten. Sie kommen häufig über Jahre mit diesen ersten zehn Befehlen aus. Denn sie leisten ja schon die wichtigsten Aufgaben:

1. Man kann sich den Inhalt der Verzeichnisse, den Inhalt der Dateien, den Aufbau von Festplatte und Disketten zeigen lassen: DIR, TYPE, TREE.

2. Man kann Disketten bereitmachen für Sicherheitskopien (Betriebsmittel-Verwaltung): FORMAT, COPY.

3. Man beherrscht das Datei-Handling, das heißt die Manipulation der Dateien im Ganzen. Man kann umbenennen, kopieren und löschen: REN, COPY, DEL.

4. Man beherrscht das Handling der Verzeichnisse und Pfade - kurz: die Organisation der externen Speicher: MKDIR (MD), CHDIR (CD), RMDIR (RM).

Die Befehle des folgenden Abschnitts vertiefen diese Kenntnisse - also das Zeigenlassen, die Betriebsmittel-Verwaltung, das Datei-Handling und die Speicher-Organisation.

15 Befehlsmodifikationen im Protected-Mode und weitere zwölf Befehle für den fortgeschrittenen Bediener

Wir haben bis jetzt die zehn wichtigsten Befehle und den einfachen more-Zusatz erläutert. Sie sind unerlässlich, will man mit dem neuen Betriebssystem arbeiten. Alle zehn Befehle auf dieser Ebene der Bedienung sind bezüglich MS-DOS - bis auf den angeführten zusätzlichen Komfort im Protected-Mode - gleich geblieben.

Was hat sich also geändert?

An drei Stellen ist die Arbeit mit bekannten Befehlen verbessert worden. Es sind alles Verbesserungen im Protected-Mode, die ihren Sinn darin haben, den Multitasking-Betrieb besser nutzen zu können. Sie betreffen also alle den neuen Betriebssystemkern cmd.exe.

Cmd.exe ist der Befehlsinterpreter des Protected-Mode. Gegenüber dem Befehlsinterpreter des Real-Mode, command.com, verfügt cmd.exe über einige Varianten in der Befehlssprache, mit der Sie kompliziertere Befehle oder Befehlskombinationen aufbauen können.

Hier die drei Punkte noch einmal kompakt und vollständig:

1. Der Wirkungsbereich einiger Befehle wurde erweitert. Es können nun Listen von Objekten, die bearbeitet werden sollen, angegeben werden. Zum Beispiel legt der folgende Befehl gleich mehrere Verzeichnisse an für Bediener desselben Personal-Computers an.

md Meier Müller Schulze Schmitz

Oder

dir Meier Müller Schulze Schmitz

zeigt den Inhalt aller vier Abteilungen, die genannt sind.

Diese Erweiterung wurde bei den folgenden Befehlen vorgenommen:

dir, mkdir, del, type, rmdir, vol.

Mit dieser neuen Möglichkeit wirken sich diese Befehle auf mehr Verzeichnisse, Dateien oder Datenträger aus. Das Letztere betrifft den vol-Befehl, der noch behandelt wird. Man kann nun also kompliziertere und vor allem länger dauernde Prozesse mit diesen Befehlen starten. Für diese Prozesse ist es dann sinnvoll, sie auf einem eigenen OS2-Command-

Prompt im Hintergrund von anderen Bildschirmgruppen ablaufen zu lassen.

2. Der Gebrauch der Wildcard-Symbole, * und ?, wurde erweitert. Sie können zum Beispiel mit dem type-Befehl nun eine ganze Familie von Dateien durchsehen.

```
type *.txt | more
```

erlaubt nun, alle Textdateien, wie sie zum Beispiel in Microsoft-Word angelegt werden, durchzublättern. In Kombination mit dem more-Befehl erhält man so einen Prozeß, der für den Multitasking-Betrieb im selben Sinne wie unter dem 1.Punkt gut geeignet ist.

3. Am more-Befehl haben wir die grundsätzliche Möglichkeit gesehen, zwei Befehle zu einem Kommando zu verknüpfen, so daß damit ein Prozeß gestartet werden kann. Das gab es insbesondere für den Pipe-Operator | schon unter MS-DOS.

MS-OS/2 eröffnet Ihnen nun eine Reihe von neuen Möglichkeiten, um Befehle zu verknüpfen. Solche Verknüpfungen werden über Operatoren ausgeführt. Zum Beispiel wurde der Trennoperator "&" eingeführt, mit dem Sie nun eine Folge von mehreren Befehlen in nur einer Befehlszeile angeben können.

Ziel: Sie wollen Verzeichnis test wieder von Ihrer Festplatte entfernen. Dazu müssen Sie zunächst alle Dateien löschen, die sich in diesem Verzeichnis befinden. Danach können Sie das Verzeichnis mit dem rd-Befehl entfernen. Im Protected-Mode geben Sie die folgende Befehlszeile ein:

```
Weg: del \test\*. * & rd test
```

Ziel: Sie wollen ein Verzeichnis für ein Projektplanungsprogramm anlegen und dann sofort die Pro-

gramm-Dateien von einer 1,2-Megabyte-Diskette auf die Festplatte laden.

```
Weg: md proj & cd proj & copy a:*. * /v
```

Das ist natürlich wieder für den Multitasking-Betrieb interessant. Denn auf diese Weise kann man schon auf der Ebene der direkten Befehlseingabe Prozesse aufrufen, die länger dauern. Es lohnt dann, diese im Hintergrund parallel zu anderen Anwendungen ablaufen zu lassen. Im Real-Mode müssen Sie die Befehle jedoch wie bisher getrennt eingeben.

Über das genannte Beispiel hinaus bietet Ihnen die Befehlssprache des cmd.exe noch weitere Operatoren, mit denen Sie komplexe Befehlsstrukturen aufbauen können. Das sind die Operatoren UND (dargestellt durch das Symbol &&) und ODER (dargestellt durch das Symbol ||). Mit den runden Klammern () können Sie nun auch Befehle zu einer Gruppe zusammenfassen.

Die folgende Tabelle zeigt eine Übersicht der alten (command.com) und neuen (cmd.exe) Befehlsoperatoren. Diese Operatoren besitzen unterschiedliche Prioritäten, so wie auch die Multiplikation bei den Rechenoperationen eine höhere Priorität besitzt als die Addition. Die Operatoren sind hier nach ihrer Priorität geordnet: (Bild 15.1)

Der Befehlsinterpreter cmd.exe führt Befehlszeilen unter MS-OS/2 auf dieselbe Art aus, wie command.com unter MS-DOS 3.x, mit Ausnahme von Befehlsfolgen, deren Ausführung durch Operatoren gesteuert wird. Das Verhalten dieser Befehlsfolgen soll im folgenden erläutert werden. Dazu geben wir jeweils den Algorithmus (das heißt die Vorschrift - Schritt für Schritt) an, nach der cmd.exe diese Befehlsfolgen ausführt. Die Ausführung der einzelnen Befehle bleibt unverändert, aber durch die Bildung

command.com	cmd.exe	Funktion
	()	Befehlsgruppenoperator
<	<	Umleitungsoperator
>	>	Umleitungsoperator
>>	>>	Umleitungsoperator
		Pipe-Operator
	&&	UND-Operator
		ODER-Operator
	&	Trennoperator für Befehle

Bild 15.1

von Befehlsgruppen und in Abhängigkeit von der erfolgreichen Ausführung von Einzelbefehlen kann die Befehlssequenz durch die folgenden Operatoren gesteuert werden.

Für eine ausführliche Darstellung aller Möglichkeiten, die man mit diesen Operatoren hat, ist hier nicht der Platz. Das ist kein Einstiegsthema mehr. Aber für den einfachen Gebrauch wollen wir doch die Regeln für deren Verwendung nennen und ein einfaches Beispiel geben. Wenn Sie schon etwas Übung im Gebrauch von MS-OS/2 haben, dann stellen sich kompliziertere und dann auch effektivere Anwendungen der Operatoren von selbst ein.

Der UND-Operator "&&"

Der Algorithmus, das ist der schrittweise Ablauf, für die UND-Operation lautet wie folgt:

1. Schritt: Führe die Anweisung auf der linken Seite des Operators aus.
2. Schritt: Wenn ein Fehler auftritt, dann brich die Bearbeitung ab und melde den Fehler zurück.
3. Schritt: Ansonsten führe die Anweisung auf der rechten Seite des Operators aus.

Beispiel:

Ziel: Sie wollen mit einem Befehl eine Diskette formatieren und zugleich die Dateien des Unterverzeichnisses test auf die neu formatierte Diskette kopieren. Dazu geben Sie den folgenden Befehl ein:

Weg: `format a:/v && copy c:\test a:/v`

Der ODER-Operator "||"

Der Algorithmus für die ODER-Operation lautet folgendermaßen:

1. Schritt: Führe die Anweisung auf der linken Seite des Operators aus.
2. Schritt: Wenn die Ausführung erfolgreich ist, melde die erfolgreiche Ausführung zurück.
3. Schritt: Ansonsten führe die Anweisung auf der rechten Seite des Operators aus und melde das Ergebnis zurück.

Beispiel:

Ziel: Sie wollen von Ihrem Wurzelverzeichnis ein COPY auf eine Diskette im Laufwerk A: machen. Sie wissen aber nicht, ob Ihre Diskette schon formatiert ist. Das können Sie mit einem ODER regeln: Sie haben formatierte oder unformatierte Disketten. Also geben Sie für die beiden möglichen Fälle die folgende Befehlssequenz ein:

Weg: `copy c:\ a: || (format a:/v:os2wurzel && copy c:\ a:/v)`

Der erste Befehl links vom ODER versucht ein copy ohne vorhergehende Formatierung. Wenn dieser Versuch erfolgreich ist, das heißt wenn die Diskette schon formatiert ist, dann endet hiermit die Anweisung. Im Falle einer nichtformatierten Diskette gibt der Befehl links vom ODER eine Fehlermeldung. Das bewirkt, daß der Befehl rechts vom ODER ausgeführt wird: erst Formatieren, dann Kopieren. Wir haben hier nur der Deutlichkeit halber Klammern verwendet, grundsätzlich sind sie hier nicht notwendig, weil UND höherrangig als ODER ist.

Der Trennoperator für Befehle "&"

Der Algorithmus zur Bearbeitung des Trennoperators lautet wie folgt:

1. Schritt: Führe die Anweisung auf der linken Seite des Operators aus.
2. Schritt: Führe die Anweisung auf der rechten Seite des Trennoperators aus und melde das Ergebnis dieser zweiten Anweisung zurück.

Beispiel:

Ziel: Sie haben eine neue Diskette mit irgendwelchen Anwendungen erhalten. Sie wollen sich umfassend über den Status dieser Diskette informieren. Das können Sie nun mit einer einzigen Befehlsfolge erledigen:

Weg: `dir A: | more & tree a:/f | more & type A:*.1st | more`

Zuerst werden die Dateien oder Verzeichnisse auf der Diskette gezeigt, dann die Struktur im Aufbau der Verzeichnisse, falls vorhanden, und schließlich die Datei readme.1st, von der man ausgehen kann,

daß sie mit diesem oder einem ähnlichen Vornamen existiert. Alle Befehle haben den Zusatz more, der vor dem Verkettungsoperator vorrangig behandelt wird und das Durchblättern erlaubt.

Weitere zwölf Befehle für den fortgeschrittenen Gebrauch

Sie haben im 14.Abschnitt die erste Gruppe von Befehlen kennengelernt, mit denen Sie in Dateien und in die Datei-Verwaltung Einblick nehmen können und die es Ihnen erlaubt, einige Manipulationen mit Dateien vorzunehmen. Sodann haben Sie nun die Neuerungen im Gebrauch altbekannter Befehle erfahren. Bei den folgenden Befehlen geht es nun darum, daß Sie die notwendigsten Befehle für einen zwar seltenen, aber dafür fortgeschrittenen Gebrauch kennenlernen. Es handelt sich um zwei Gruppen von Befehlen: Die eine Gruppe erweitert Ihr Bedürfnis, Einsicht in das System zu nehmen und die zweite Gruppe dient zur Pflege und Organisation Ihrer eigenen Daten.

Neben dir, tree und type stellen helpmsg, date, time, vol, ver, chkdsk und path weitere Möglichkeiten dar, sich Informationen über den Zustand des Systems und der externen Speicher zu verschaffen. Dabei können zumeist auch neue Werte gesetzt werden.

Der erste Befehl im Bunde dieser elementaren Zeigebefehle ist ein Hilfe-Befehl, der ebenfalls in MS-OS/2 neu ist:

Funktion: HELPMMSG liefert erklärende Informationen über eine Warnung oder Fehlermeldung des Systems.

Wo: Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

helpmsg Meldungsnummer

Bemerkung: Das helpmsg-Kommando liefert Erläuterungen zu den MS-OS/2-Meldungen und zusätzliche Informationen darüber, wie darauf zu reagieren ist.

Im Protected-Mode besitzt jede MS-OS/2-Meldung eine Meldungskennung der Form "DOSnnnn", wobei nnnn eine vierstellige Meldungsnummer ist. Beachten Sie bitte, daß im Real-Mode im allgemeinen keine Meldungskennung ausgegeben wird. Um mehr Informationen über eine MS-OS/2-Meldung zu erhalten, müssen Sie das Kommando helpmsg gefolgt von der entsprechenden Meldungsnummer eingeben.

Als Antwort wird angezeigt:

1. die Meldung selbst, wobei die darin enthaltenen Variablen von MS-OS/2 durch "****" ersetzt werden können,
2. gewisse Erläuterungen zur Bedeutung der Meldung,
3. ein Vorschlag, wie Sie auf die Meldung reagieren können.

Ziel: Sie wollen die Meldung zur Nr. 0002 einsehen.

Weg: helpmsg 0002

oder

Weg: helpmsg 2

Resultat: die Ausgabe erscheint am Bildschirm: (Bild 15.2)

Funktion: DATE zeigt das Datum und erlaubt die Eingabe eines neuen bzw. die Veränderung des dem System bekannten Datums.

DOS0002: The system cannot find the file specified.

EXPLANATION: The file named in the command does not exist in the directory or the drive specified. You may have typed the filename incorrectly, or you may have used an unacceptable character. unacceptable characters are: . " / \ [] : ; < > + , and all control characters.

ACTION: Retry the command using a correct filename.

Bild 15.2

Anwendungen für Fortgeschrittene

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

date [mm-tt-jj]

Ziel: Mit dem folgenden Befehl setzen Sie das Datum auf den 21. Juli 1988:

Weg: date 21-7-88

Bemerkung: Das Datum kann mit dem date-Kommando direkt eingegeben werden. Als Trennzeichen sind Bindestriche (-), Punkte (.) und Schrägstriche (/) erlaubt. Für die Zahlen mm, tt und jj sind folgende Wertebereiche (Zahleingaben) zulässig, anderes wird abgewiesen:

mm: 1 bis 12

tt: 1 bis 31

jj: 80 bis 99 oder 1980 bis 2079

Wenn Sie das date-Kommando ohne Datumsangabe eingeben, so wird das aktuelle Datum gezeigt und ein neues Datum angefordert. Soll das Datum nicht geändert werden, genügt es, die Eingabetaste zu betätigen.

Das country-Kommando in der Datei config.sys gestattet eine Änderung des Eingabeformats zu tt-mm-jj, falls dort der entsprechende Landesschlüssel ausgewiesen wurde.

Funktion: TIME erlaubt die Eingabe oder die Änderung der Systemuhrzeit.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

time [Stunden:Minuten[:Sekunden[.Hundertstel]]]

Ziel: Sie wollen die Uhrzeit des Systems auf 17.54 Uhr setzen. Das können Sie mit diesem Befehl tun:

Weg: time 17:54

Bemerkung: Ähnlich wie mit dem date-Befehl kann man die Zeit-Festsetzungen manipulieren. MS-OS/2 benutzt für die Uhrzeit ein 24-Stunden-Format. Die Zeitinformation wird bei der Aktualisierung eines Verzeichnisses verwendet, wenn Dateien erstellt oder verändert werden.

Die Eingabe von time ohne Optionen zeigt die aktuelle Zeit an und ermöglicht eine Änderung. Soll die Zeit nicht geändert werden, ist lediglich die Eingabetaste zu betätigen wie beim Datum.

Die Angabe eines neuen Wertes muß im 24-Stunden-Format erfolgen. Dabei sind die folgenden Wertebereiche für Zahleingaben zu beachten:

Stunden: 0 bis 23

Minuten: 0 bis 59

Sekunden: 0 bis 59

Hundertstel: 0 bis 99

Die Trennzeichen müssen Sie dabei der Zeitanzeige entsprechend wählen. Die neue Zeitangabe können Sie auch direkt in der Kommandozeile mitangeben. Wenn Sie die Zeitangabe nicht korrekt getippt haben, wird eine erneute Eingabe angefordert.

Funktion: VER zeigt die Nummer der OS/2-Version an.

WO: im Real-Mode und im Protected-Mode

Syntax: im Real-Mode und Protected-Mode:

ver

Bemerkung: Die Eingabe von ver liefert die Nummer der benutzten OS/2-Version zurück.

Ziel: Sie wollen die Version von MS-OS/2 feststellen, die Sie beim Kauf Ihres Personal-Computers erhalten haben

Weg: ver

Resultat: Es erscheint die Meldung

The Microsoft Operation System/2 Version is 1.00

Funktion: VOL zeigt den vom Benutzer gewählten Namen einer Diskette oder Festplatte an.

WO: im Real-Mode und im Protected-Mode

Syntax im Real-Mode:

vol [d:]

Syntax im Protected-Mode:

vol [d:] [...]

Ziel: Sie wollen den Namen der Festplatte, der vom "Installateur" gewählt wurde, anzeigen lassen.

Weg: vol C:

Resultat: Es erscheint die Meldung

Volume in drive C has no label

(Die Festplatte C besitzt keinen Namen)

Bemerkung: Wird kein Laufwerk angegeben, so wird das Standardlaufwerk angenommen. Weitere Informationen über die vom Benutzer zu vergebenden Namen können bei den Kommandos label und format im Handbuch nachgeschlagen werden. Es sei hier nur angemerkt, daß man mit dem label-Befehl auch nachträglich noch Namen vergeben kann.

NEU im Protected-Mode: Sie können nun mehrere Laufwerksnamen in einer Zeile angegeben werden. Die Namen werden dann in der angegebenen Reihenfolge angezeigt.

Funktion: PATH definiert einen Suchpfad für externe Kommandos.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

path [d:] [Pfad] [] ...]

oder

path ;

Ziel: Sie wollen den Pfad ausschließlich in die Wurzel, die Verzeichnisse \os2\bin, os2\rbin und rbsys legen.

Weg: path c:\;c:\os2\bin;c:\os2\rbin;c:\rbsys

Bemerkung: Wenn man mit mkdir und chdir eine komplexe und verzweigte Struktur der Speicher-Verwaltung und Arbeitsweise mit Dateien auf seinem Rechner installiert, dann ist es von besonderer Wichtigkeit, allgemein verbindliche Wege oder Pfade fest-

zulegen, denen bei einem Arbeitsvorgang nachgegangen wird. Mit dem path-Befehl (Pfad-Befehl) können Sie solche Wege festschreiben. Auf diesem Weg wird dann stets ein externer Befehl gesucht. Für nicht-ausführbare Dateien (zum Beispiel ASCII-Textdateien oder sonstige Daten-Dateien) können Sie ebenfalls einen Suchpfad definieren. Dazu müssen Sie die Befehle append (Real-Mode) oder dpath (Protected-Mode) verwenden. Das sei aber hier nur angemerkt.

Das path-Kommando teilt dem System mit, in welchen Verzeichnissen nach externen Kommandos oder Programmen gesucht werden soll, wenn diese nicht im aktuellen Verzeichnis gefunden werden. Externe Kommandos bzw. Befehle sind solche, die außerhalb des jeweiligen Befehlsinterpreters liegen: command.com für den Real-Mode und cmd.exe für den Protected-Mode. Ein Suchpfad ist solange gültig, bis ein neuer Pfad angelegt wird.

MS-OS/2 sucht die Pfade grundsätzlich in der Reihenfolge ab, in der sie im path-Kommando angegeben werden. Wenn Sie ein externes Kommando eingeben, das in mehreren Verzeichnissen existiert, so wird zur Ausführung die erste Kopie verwendet, die im Such-Pfad gefunden wird.

Ziel: Sie wollen sehen, ob ihr Pfad korrekt angenommen wurde.

Weg: path

Wenn Sie das path-Kommando ohne Optionen eingeben, wird der aktuelle Pfad angezeigt. Der Pfad-Befehl erzeugt also nicht nur Wege, sondern zeigt sie auch unter dieser Bedingung an. Mehrere Stationen auf dem Such-Pfad werden mit einem Semikolon getrennt.

Ziel: Sie wollen den aktuellen Pfad löschen. Geben Sie ein:

Weg: path ;

Das wirkt sich dann so aus, daß MS-OS/2 nur noch im aktuellen Verzeichnis nach externen Kommandos sucht.

Falls MS-OS/2 mit dem Install-Programm installiert wird, wird das path-Kommando automatisch in die Dateien autoexec.bat und startup.cmd eingefügt.


```
C:\> chkdsk b:

362496 bytes total disk space.
38912 bytes in 2 hidden files.
26624 bytes in 3 user files.
296960 bytes available on disk.
[DOS mode storage report]
654336 bytes total storage.
5400024 bytes free.
```

Bild 15.3

Funktion: CHKDSK überprüft das Inhaltsverzeichnis einer Diskette auf vorhandene Fehler.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

```
chkdsk [d:] [Pfad] [Dateiname] [/f] [/v]
```

Bemerkung: Wenn das chkdsk-Kommando Fehler auf dem Datenträger entdeckt, dann werden entsprechende Fehlermeldungen ausgegeben. Wenn nicht, dann wird der Status des Datenträgers im angegebenen Laufwerk ausgegeben. Im Real-Mode wird Ihnen zusätzlich der Status des Hauptspeichers angezeigt.

Ziel: Sie wollen den Status der Diskette in Laufwerk B abrufen. Sie tun das vom Real-Mode aus.

Weg: chkdsk b:

Resultat: MS-OS/2 meldet (Bild 15.3)

Vom Protected-Mode aus liefert der Befehl eine leicht veränderte Meldung.

Ziel: Sie wollen die Festplatte vom Protected-Mode aus untersuchen lassen.

Weg: chkdsk c: (Bild 15.4)

Bemerkung: Es fehlt bei der Protected-Mode Meldung die Angabe der Größe des Arbeitsspeichers. Diese Angabe hätte hier auch keinen Sinn, weil MS-OS/2 im Protected-Mode einen virtuellen Arbeitsspeicher adressiert.

Parameter: Wenn Sie den Parameter /f hinzufügen, werden die im Inhaltsverzeichnis des Laufwerks gefundenen Fehler automatisch korrigiert. Beim parallelen Ablauf von Hintergrundprozessen im Protected-Mode sollten Sie jedoch auf die Angabe des Parameters /f verzichten.

Wenn Sie den Parameter /v angeben, dann werden während der Abarbeitung des Kommandos laufend Nachrichten über den Stand der Verarbeitung angezeigt.

Falls ein Pfad mit Dateiname spezifiziert wird, so wird der Status für das Laufwerk und für die angegebene Datei berichtet.

So ergänzt der Check-the-disk-Befehl die Informationen, die schon die Befehle dir und type liefern. Zusammen mit den Zeige-Befehlen helpmsg, path, date, time, ver und vol sind die wichtigsten nun angegeben. Helpmsg, date, time und ver beziehen sich nicht auf die Datenträger, sondern auf das System selbst, sofern Sie unter anderem in Ihrem Rechner eine Uhr installiert haben.

```
[c:\] chkdsk c:

33462272 bytes total disk space.
321536 bytes in 4 hidden files.
69632 bytes in 28 directories.
21233664 bytes in 773 user files.
61440 bytes in bad sectors.
11776000 bytes free.
```

Bild 15.4

Wir kommen nun zum erweiterten Datei-Handling. Am Anfang soll der Vergleich von Dateien stehen.

Wenn Sie zum Beispiel beim copy-Befehl den Parameter /v vergessen haben und nachträglich prüfen wollen, ob die Daten tatsächlich richtig kopiert wurden, dann können Sie die Dateien vergleichen (compare). Natürlich können Sie das auch in jeder anderen Situation tun.

Funktion: COMP vergleicht zwei Dateien oder Dateimengen.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

`comp [d:] [Pfad] [Dateiname] [d:] [Pfad]`

Ziel: Sie wollen überprüfen, ob Ihre Dateien im Verzeichnis test mit den gleichnamigen Dateien in der Wurzel übereinstimmen. Diesen Vergleich führen Sie mit dem folgenden Befehl durch:

Weg: `comp c:\test c:\`

Bemerkung: Das comp-Kommando vergleicht zwei Dateien oder Dateimengen miteinander, die jeweils durch eine Laufwerksangabe, einen Pfad und einen Dateinamen spezifiziert werden.

Die Dateimengen können sich in verschiedenen Verzeichnissen oder auch auf verschiedenen Laufwerken befinden.

Falls die Dateimengen, die verglichen werden sollen, denselben Pfad und dieselben Dateinamen haben - vorausgesetzt sie befinden sich auf verschiedenen Laufwerken - so kann bei der Angabe der zweiten Dateimenge der Pfadname entfallen.

Bei der Spezifikation der Dateinamen können Sie die Wildcards * und ? verwenden. Wenn Sie statt eines Pfades mit Dateinamen nur einen Pfad angeben, so nimmt MS-OS/2 *.* als Dateiname an.

Wenn Sie den zweiten Pfad (ggf. mit Dateiname) nicht angeben, so wird er durch das System angefordert. Comp kann auch gänzlich ohne Parameter eingegeben werden. Das System fordert dann die benötigten Informationen einzeln an.

Während des Vergleichs werden die Namen der verglichenen Dateien und eine Meldung, ob die Dateien übereinstimmen, ausgegeben. Falls zwei miteinander verglichene Bytes verschiedene Inhalte aufweisen, so erscheint eine Fehlermeldung. Diese Fehlermeldung zeigt den Offset und die Inhalte der beiden Bytes in hexadezimaler Notation an. Unter dem Offset versteht man die relative Position eines Bytes zum Dateianfang.

Nach der zehnten Fehlermeldung wird die comp-Prozedur abgebrochen.

Dateien unterschiedlicher Länge können bis zur Länge der kleineren Datei verglichen werden. Das System fragt jedoch vorher an, ob der Vergleich gewünscht wird.

Wenn es keine weiteren Dateien mehr gibt, die mit dem ersten Pfad mit Dateinamen übereinstimmen, wird angefragt, ob weitere Dateien verglichen werden sollen. Falls Sie mit J antworten, werden zwei neue Pfade (ggf. mit Dateinamen) angefordert.

Funktion: DISKCOMP vergleicht den Inhalt der Quelldiskette im ersten angegebenen Laufwerk mit dem Inhalt der Zieldiskette im zweiten angegebenen Laufwerk.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

`diskcomp [d1:] [d2:]`

Ziel: Sie wollen den Inhalt der Disketten in Laufwerk A und Laufwerk B miteinander vergleichen. Sie können dies mit dem folgenden Befehl tun:

Weg: `diskcomp a: b:`

Bemerkung: So wie dem copy-Befehl der comp-Befehl zugeordnet werden kann, so gehören diskcomp (Disk-compare!) und der diskcopy-Befehl, der als nächster erklärt wird, zusammen.

Wenn Sie dasselbe Laufwerk als Quelle und als Ziel der Operation angeben, dann wird wie beim diskcopy-Vorgang vom System jeweils das Einlegen der Quellen- bzw. der Zieldiskette verlangt.

Der Vergleich der Disketten erfolgt spurweise. Stimmt das Format der Disketten nicht überein, so wird die Routine mit einer Fehlermeldung abgebrochen.

Falls zwei Spuren gefunden werden, die ungleich sind, dann erhalten Sie eine Fehlermeldung, die Angaben über Spur und Seite des Fehlers enthält.

Nachdem der Vergleich abgeschlossen wurde, fragt das System an, ob Sie weitere Disketten vergleichen wollen.

Das diskcomp-Kommando arbeitet nicht mit Laufwerken, die durch die Befehle assign, join oder subst anders zugewiesen wurden. Wenn Sie solche Laufwerke angeben, erhalten Sie eine Fehlermeldung.

Funktion: DISKCOPY kopiert den Inhalt der Queldiskette im ersten angegebenen Laufwerk auf die Zieldiskette im zweiten angegebenen Laufwerk.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

diskcopy [d1:] [d2:]

Ziel: Sie wollen den gesamten Inhalt der Diskette in Laufwerk A auf die Diskette in Laufwerk B kopieren. Das erledigen Sie wie folgt:

Weg: diskcopy a: b:

Bemerkungen: Für Sicherheitskopien - oder allgemein: für Duplikate von Disketten steht Ihnen der diskcopy-Befehl (Disketten-Kopierer) zur Verfügung. Er erfüllt im Grunde zwei Aufgaben: Es wird beim Kopieren zugleich formatiert. Er ersetzt also den format- und den copy-Befehl. Allerdings können Sie diese Vorteile nur dann nutzen, wenn Sie komplette Disketten duplizieren. MS-OS/2 schreibt alle Dateien der Ursprungs- bzw. Queldiskette an die physikalisch gleiche Stelle auf der Zieldiskette.

Warnung: Die Dateien der Zieldiskette, falls welche vorhanden sind, werden durch das diskcopy-Kommando zerstört.

Nachdem Sie die Routine Diskcopy aufgerufen haben, meldet sich MS-OS/2 mit der Aufforderung, die Quell- und die Zieldiskette in die entsprechen-

den Laufwerke einzulegen. Nach Betätigung einer beliebigen Taste wird mit dem Kopiervorgang begonnen.

Wenn MS-OS/2 auf einer Diskette Fehler entdeckt, so werden diese Fehler gemeldet und der Kopiervorgang weiter durchgeführt.

Nachdem der Kopiervorgang beendet ist, fragt das System an, ob weitere Disketten kopiert werden sollen. Die folgenden Kopiervorgänge werden von demselben Quelllaufwerk aus auf dasselbe Ziellaufwerk durchgeführt.

Wenn Sie nur ein Laufwerk am Rechner zur Verfügung haben, dann geben Sie als Quelle und Ziel dasselbe Laufwerk an. Das diskcopy-Kommando wird dann über das eine Einzellaufwerk abgewickelt. Dabei fordert das System jeweils zum Einlegen von Quellen- bzw. Zieldiskette auf.

d1 und d2 können also auch dasselbe Laufwerk angeben. Wenn keine Laufwerke genannt werden, so werden die Laufwerksnamen angefordert. Eine formatierte oder unformatierte Zieldiskette wird dem Format der Quellendiskette entsprechend formatiert.

Falls die Laufwerkstypen oder die Diskettentypen verschieden sind, wird eine entsprechende Fehlermeldung gezeigt. Das kann zum Beispiel vorkommen, wenn Sie ein 1,2 Megabyte und ein 360 Kilobyte-Laufwerk zur Verfügung haben.

Bemerkung für Kenner: Der diskcopy-Befehl arbeitet nicht auf Laufwerken, die in subst- oder join-Kommandos verwendet wurden. Ebenso werden Laufwerkszuweisungen mit dem assign-Kommando ignoriert.

Neben dem diskcopy-Befehl und seiner besonderen Funktion ergänzt auch der folgende Kopierbefehl das copy-Kommando.

Funktion: XCOPY kopiert Dateien und Verzeichnisse, sowie deren Unterverzeichnisse, falls solche existieren.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

xcopy [d:] [Pfad] [[d:] [Pfad]] [/s] [/e] [/p] [/v] [/a] [/m] [/d:tt-mm-jj]

Ziel: Mit dem folgenden Befehl wollen Sie alle Dateien und Unterverzeichnisse von der Diskette in Laufwerk A auf die Diskette in Laufwerk B kopieren. Dabei bleibt die Struktur des Dateibaumes komplett erhalten. Auch leere Unterverzeichnisse werden kopiert.

Weg: xcopy a: b: /s /e

Bemerkung: Neben die beiden Kopierbefehle, die wir schon erläutert haben, treten noch zwei andere. Beide lösen besondere Aufgaben, die sich bei umfangreicheren Arbeiten am Personal-Computer einstellen können. Wenn Sie unmittelbar beim Kopieren dem komplexen Aufbau Ihrer Verzeichnisse Herr werden wollen, bzw. wenn Sie überhaupt nichts davon merken wollen, dann verwenden Sie den XCOPY-Befehl!

Die erste Angabe von Laufwerk und Pfad (ggf. mit Dateiname) bezeichnet die Quelldatei oder das Quellverzeichnis, das kopiert werden soll. Dementsprechend bezieht sich die zweite Angabe auf das Ziel der Operation. Mindestens einer der Quellenparameter, Laufwerk oder Pfad, muß angegeben sein. Wenn Sie keine Angaben über das Ziel machen, so wird das aktuelle Verzeichnis als Ziel angenommen, wie das stets bei ähnlichen Befehlen der Fall ist.

Wenn Sie kein Laufwerk, keinen Pfad und auch keinen Dateinamen oder Dateinentyp (mit Wildcards) angeben, dann wird das aktuelle Verzeichnis und der Dateiname *.* angenommen.

Der Parameter /s kopiert Verzeichnisse und Unterverzeichnisse - aber nur, sofern diese nicht leer sind. Wenn Sie diesen Parameter nicht angeben, arbeitet xcopy normal innerhalb eines Verzeichnisses.

Der Parameter /e kopiert Unterverzeichnisse auch dann, wenn sie leer sind. Der Parameter /e darf nicht zusammen mit /s verwendet werden.

Der Parameter /p fragt für jede Zieldatei nach, ob sie kopiert werden soll: "(Y/N)".

Der Parameter /v bewirkt, daß MS-OS/2 überprüft, ob die erzeugten Zieldateien mit den Quelldateien identisch sind. Das ist das übliche "Verify" beim Kopieren, wie wir es schon kennen.

Der Parameter /a kopiert diejenigen Quelldateien, deren Archivbit gesetzt ist. Dabei wird das Archivbit der Quelldateien nicht verändert. Mit Hilfe des Archivbits können Sie bestimmte Dateien oder Dateigruppen markieren. Weitere Informationen über das Archivbit können Sie beim attrib-Kommando im MS-OS/2-Handbuch nachlesen.

Der Parameter /m hat die gleiche Wirkung wie der Parameter /a, jedoch werden die Archivbits der Quelldatei gelöscht.

Der Parameter /d:tt-mm-jj läßt Quelldateien kopieren, die an oder nach einem bestimmten Datum modifiziert wurden. Das Format der Datumsangabe kann je nach benutztem Landesschlüssel abweichen. Vergleichen Sie dazu auch das date-Kommando.

Das xcopy-Kommando sollten Sie anstelle von diskcopy verwenden, wenn sich die Quelldateien in verschiedenen Unterverzeichnissen befinden oder wenn sogar unterschiedliche Diskettenformate vorliegen.

Funktion: BACKUP sichert eine oder mehrere Dateien von einem Laufwerk auf ein anderes.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

```
backup [d:] [Pfad] [Dateiname] [d:] [/s] [/m] [/f]
[/d:Datum] [/t:Zeit] [/L:Dateiname]
```

Bemerkung: Der letzte hier zu verzeichnende Kopierbefehl löst insbesondere die Aufgabe, Dateien, die größer sind als die Speicher-Kapazität einer Diskette, dennoch zu sichern. Die backup-Routine reißt auch Dateien auseinander und protokolliert dies für die spätere Restaurierung mit dem restore-Befehl.

Es spielt keine Rolle, ob es sich bei den Laufwerken um ein Diskettenlaufwerk oder ein Festplattenlaufwerk handelt. Alle Kombinationen sind möglich. Auch das Format der Disketten braucht nicht übereinzustimmen.

Auf dem ersten angegebenen Laufwerk befinden sich die zu sichernden Dateien. Das zweite Laufwerk ist das Backup-Laufwerk.

Ziel: Sie wollen eine vollständige Sicherheitskopie von Ihrer Festplatte erstellen. Das machen Sie mit dem folgenden Befehl:

Weg: backup c:\ a: /s

Parameter: Wenn Sie den Parameter "/a" nicht verwenden, dann werden zunächst alle Dateien auf der Backup-Diskette gelöscht, bevor die neuen Dateien kopiert werden. Die einzelnen Parameter haben die folgenden Bedeutungen:

Der Parameter /s bewirkt, daß Unterverzeichnisse ebenfalls gesichert werden.

Der Parameter /m bewirkt, daß nur diejenigen Dateien gesichert werden, die seit dem letzten Sicherungsvorgang verändert wurden.

Der Parameter /a bewirkt, daß die zu sichernden Dateien zu den bereits auf der Sicherungsdiskette vorhandenen hinzugefügt werden. Die alten Dateien auf dieser Sicherungsdiskette werden nicht gelöscht. Dies gilt jedoch nicht für solche Sicherungsdisketten, die unter MS-DOS 3.2 oder früheren DOS-Versionen erstellt wurden. Dieser Parameter ist neu im MS-DOS/2.

Der Parameter /f bewirkt, daß eine unformatierte Zieldiskette beim Backup-Vorgang gleichzeitig formatiert wird.

Der Parameter /d:Datum bewirkt, daß nur die Dateien gesichert werden, die am angegebenen Tag oder später verändert wurden.

Der Parameter /t:Zeit bewirkt, daß nur die Dateien gesichert werden, die zur angegebenen Zeit oder später verändert wurden.

Der Parameter /L:Dateiname erzeugt eine Datei im Wurzelverzeichnis desjenigen Laufwerks, das die zu sichernden Dateien enthält. Fehlt der Dateiname, so wird die Datei backup.log erzeugt. Diese Datei ist wie folgt aufgebaut:

Die erste Zeile enthält Datum und Zeit des Sicherungsvorgangs. Die folgenden Zeilen beziehen sich auf je eine gesicherte Datei und enthalten jeweils den Dateinamen und die Nummer der Sicherungsdiskette, auf der sich die Sicherheitskopie befindet. Diese Information ist unerlässlich, falls eine spezielle

Datei restauriert werden soll, denn nur über die Nummer der Sicherungsdiskette ist die globale Adresse der Datei bekannt. An eine bereits existierende backup.log-Datei werden die neuen Eintragungen angehängt.

Bemerkung für Kenner: Wird das zu sichernde Laufwerk im Zusammenhang mit den Kommandos assign, join oder subst verwendet, sollten Sie das backup-Kommando nicht benutzen. Dann können die Dateien unter Umständen mit dem restore-Kommando nicht mehr restauriert werden.

Funktion: RESTORE speichert Dateien zurück, die mit dem backup-Kommando gesichert wurden.

Wo: im Real-Mode und Protected-Mode

Syntax im Real und Protected-Mode:

restore d1: [d2:] [Pfad] [Dateiname] [/s] [/p]
[/b:Datum] [/a:Datum] [/e:Zeit] [/L:Zeit] [/m] [/n]

Bemerkung: Die Komplementär-Routine zum backup ist die Restaurierung, die Wiederherstellung, Ihrer Dateien oder auch Ihrer gesamten Datei-Organisation mit dem restore-Befehl. Wenn - was nicht zu wünschen ist - Ihre Festplatte gelöscht wird, dann werden Sie die Funktion von backup und restore zu schätzen wissen.

d1 enthält die gesicherten Dateien, während d2 das Ziellaufwerk bezeichnet. Pfad (ggf. mit Dateiname) bezeichnet das Verzeichnis und die Dateien, die zurückgespeichert werden sollen.

Das restore-Kommando arbeitet auch mit Laufwerken unterschiedlichen Typs.

Ziel: Sie wollen die Datei \rechnung\monat.akt von der Backup-Diskette in Laufwerk A in das Verzeichnis \rechnung auf Laufwerk C zurückspeichern. Verwenden Sie dazu den Befehl:

Weg: restore a: c:\rechnung\monat.akt

Folgende Parameter können Sie beim Restore-Befehl verwenden:

Der Parameter /s bewirkt, daß auch Unterverzeichnisse zurückgespeichert werden.

Der Parameter /p bewirkt, daß Sie bei jeder Datei, die nur gelesen werden darf oder die sich seit dem letzten Backup geändert hat, zunächst um Erlaubnis gefragt werden, ob diese Dateien zurückgespeichert werden sollen.

Der Parameter /b:Datum läßt nur die Dateien zurückspeichern, die an oder vor dem angegebenen Datum zuletzt verändert wurden.

Der Parameter /a:Datum läßt nur die Dateien zurückspeichern, die an oder nach dem angegebenen Datum zuletzt verändert wurden.

Der Parameter /e:Zeit läßt nur die Dateien zurückspeichern, die zur angegebenen Zeit oder früher zuletzt verändert wurden.

Der Parameter /l:Zeit läßt nur die Dateien zurückspeichern, die zur angegebenen Zeit oder später zuletzt verändert wurden.

Der Parameter /m läßt nur die Dateien zurückspeichern, die seit dem letzten Backup verändert wurden.

Der Parameter /n läßt nur die Dateien zurückspeichern, die auf dem Ziellaufwerk nicht mehr existieren.

Nachdem Sie zurückgespeichert haben, sollten Sie mit den Kommandos dir oder type überprüfen, ob die Datei oder die Dateien korrekt zurückgespeichert worden sind.

Achtung: Restore kann nicht zum Zurückspeichern der Systemdateien OS2bio.com, OS2dos.com, cmd.exe oder command.com verwendet werden. Um OS2bio.com und OS2dos.com zurückzuspeichern, muß das sys-Kommando verwendet werden. Command.com und cmd.exe können mit dem copy-Kommando zurückgespeichert werden.

Die hier beschriebenen Befehle für Kopieren, Vergleichen und Wiederherstellen von Dateien gestatten es Ihnen, schon recht komplexe Operationen im Zusammenhang Ihrer Dateiverwaltung durchzuführen. Sie unterscheiden nicht zwischen den beiden in MS-OS/2 vorhandenen Arbeitszuständen, Real-Mode und Protected-Mode. Sofern Sie in der MS-DOS-3.x-Umgebung bewandert sind, werden Sie sicherlich erfreut festgestellt haben, daß sich an dem Gewohnten nichts verändert hat.

16 MS-OS/2-Presentation-Manager: Die grafische Bedieneroberfläche analog zu Windows

Microsoft-Windows ist eine grafische Benutzeroberfläche für MS-DOS, die derzeit noch separat angeschafft und unter MS-DOS installiert werden muß. Für MS-OS/2 ist vorgesehen, Windows als Presentation-Manager, das heißt als integrierten Bestandteil des Betriebssystems mitauszuliefern. Dieses "OS/2-Windows" heißt MS-OS/2-Presentation-Manager.

Windows ist von vornherein von Microsoft als grafisches Interface mit Blick auf die INTEL Prozessoren 80286 und 80386 entwickelt worden. Interface bedeutet hier die Schnittstelle, gleichsam den Übersetzer zwischen dem Bediener und dem Betriebssystem einerseits und dem Anwendungsprogrammierer und dem Betriebssystem andererseits. Daß es Windows schon für den 8088/8086 gibt, hat folgende Gründe:

1. Es ging um die Vereinheitlichung von grafischen Oberflächen überhaupt - um die Bildung eines allgemeinen Standards. Ein Standard auf dem Gebiet der grafischen Oberfläche des Betriebssystems sollte die Einarbeitung in die Computerei erleichtern. Die Einführung von Personal-Computern am Arbeitsplatz hat nämlich gezeigt, daß die dabei entstehenden Kosten für Einarbeitung bei weitem die Investitionskosten von Hard- und Software übersteigen.

2. Der Standard sollte gleichermaßen für die Programmierer geschaffen werden. Eine gute grafische Oberfläche muß ja nicht mehrmals "erfunden", das heißt von Grund auf neu programmiert werden. Windows stellt in dieser Beziehung ein Werkzeug für Programmierer dar, das kompakte grafische Pro-

gramm-Module in Form von Funktionen zur Verfügung stellt - wie zum Beispiel den Zugriff auf Drop-down-(Pull-down) oder Pop-up-Menüs (Vgl. Abschnitt 10).

3. Aber auch die Portabilität von Anwendungsprogrammen spielt gerade in Bezug auf ihre grafische Oberfläche hier eine große Rolle. Portabilität bedeutete anfangs insbesondere, daß Programmierer sich nicht mehr um die unterschiedlichsten Varianten des BIOS (BASIC Input Output System) kümmern mußten. Das heißt: sie waren von da an unabhängig von einer zentralen Schnittstelle zwischen Hardware und Betriebssystem.

4. Schließlich war das Windows-Konzept stets von der Vorstellung beherrscht, daß grafische Anwendungen leicht an eine veränderte CPU, zum Beispiel vom 80286 auf den 8086 und umgekehrt, angepaßt werden können.

Der OS/2-Presentation-Manager bietet MS-OS/2 eine grafische Benutzeroberfläche, die ihrer kleinen Schwester, Windows 2.0, zum Verwechseln gleicht, aber mehr kann, weil sie sowohl den Zugriff auf den erweiterten Speicherraum als auch das Multitasking des INTEL Prozessors 80286 unterstützt.

Betrachtet man noch die dritte im Bunde der Benutzeroberflächen von Microsoft, nämlich Windows 386 für den Prozessor 80386, dann zeigen sich drei Rechner-Generationen nach außen hin auf die gleiche Weise. Die drei äußerlich gleichen Schwestern nutzen mit Hilfe verschiedener Betriebssysteme ebenso unterschiedliche Prozessoren.

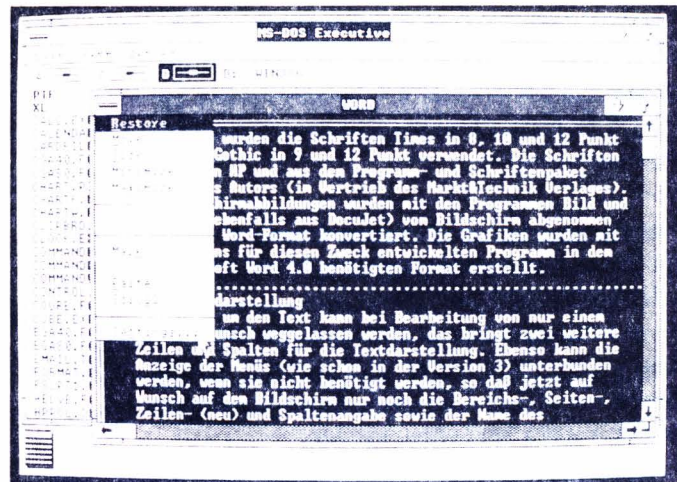


Bild 16.2

Befehlseingabe im MS-OS/2-Presentation-Manager

Der Presentation-Manager kennt ebenso wie die beiden Windows-Varianten mehrere Möglichkeiten, wie Befehle eingegeben werden. Sie können erstens den gewünschten Befehl durch "Hinzeigen auf" ein grafisches Symbol mit der Maus auswählen. Eine zweite Möglichkeit bildet die Auswahl aus einer Menüleiste ebenfalls mit der Maus. Handelt es sich dabei um ein waagerechtes Menü in der oberen Windowsleiste, dann wird ein Untermenü nach unten aufgeklappt (Pull-down-Menü), so daß der Bediener den entsprechenden Befehl durch Anklicken mit der Maus aufrufen kann. Bei komplexeren Menüs erscheint eine Dialogbox, die den Anwender auffordert, weitere Daten einzugeben, gewisse Parameter zu setzen oder eine zusätzliche Wahl zu treffen. Dialogboxen erscheinen mitten im Bild (Pop up). Die beiden Abbildungen 16.1 und 16.2 veranschaulichen die beiden Menü-Arten.

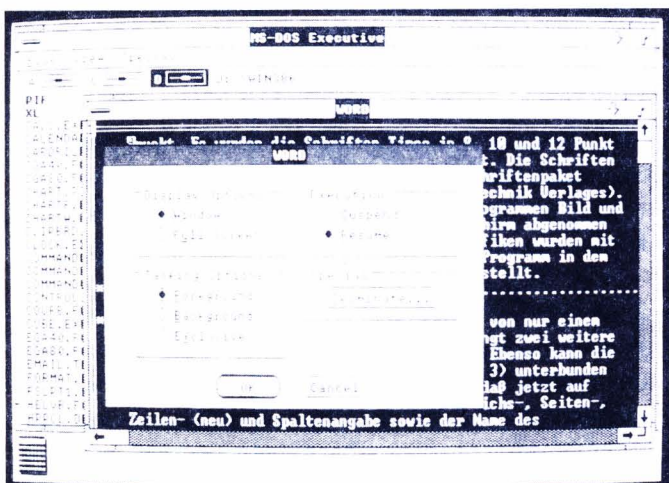


Bild 16.1

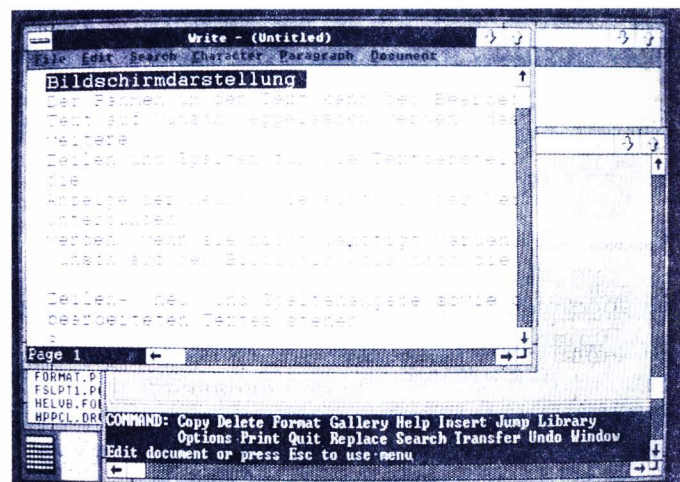


Bild 16.3

Fenstertechnik im MS-OS/2-Presentation-Manager

Wie schon unter Windows haben Sie mit dem Presentation-Manager die Möglichkeit, verschiedene Fenster auf den Bildschirm zu holen, in denen dann unterschiedliche Anwendungsprogramme laufen können. Diese Fenster können nebeneinander liegen, aber auch hintereinander. Im letzteren Fall nennt man sie auch überlappende Fenster. Die Abbildung 16.3 zeigt ein Beispiel überlappenden Fenster.

WYSIWYG-Prinzip im MS-OS/2-Presentation-Manager

Wie Windows erlaubt der OS/2-Presentation-Manager darüber hinaus einen mehr intuitiven Zugang und Zugriff auf Funktionen des Betriebssystems. Man braucht die Befehlssprache nicht mehr zu lernen, man arbeitet nur noch mit der Maus, indem man auf das hinzeigt, was man gerade braucht. Das Prinzip lautet: Was du siehst, das bekommst du auch. Auf englisch heißt das: What you see is what you get (WYSIWYG). Das erleichtert nicht nur die Einarbeitung in das neue Betriebssystem MS-OS/2, sondern macht sie sogar überflüssig für diejenigen, die Microsoft-Windows schon unter MS-DOS benutzt haben. Auch in Zukunft wird für den Bediener diese wichtige Brücke zwischen MS-OS/2 und MS-DOS nicht abgebrochen, sondern eher weitergebaut werden. Der Benutzer kann dann ohne große Probleme von einem Computer zum anderen wechseln.

Es ist durchaus denkbar und auch angestrebt, daß Software, die unter MS-DOS oder MS-OS/2 und Microsoft-Windows bzw. OS/2-Presentation-Manager entwickelt wird, auf der Oberfläche gleich aussieht. Sie kann gleichermaßen im Real-Mode laufen, sofern nur einige Einschränkungen beachtet werden. In jedem Fall unterstützen die Microsoft-Bedieneroberflächen - ob als MS-DOS-Oberfläche oder als MS-OS/2-Presentation-Manager - eine Applikationsentwicklung im Sinne dieser am Anwender orientierten Kompatibilität.

Die Vorteile, die der Anwender durch die Nutzung von MS-OS/2 unter dem Presentation-Manager hat, sind hier in fünf Punkten zusammengefaßt:

1. Die einfache Handhabung: Grafische Benutzeroberflächen sind schneller zu bedienen als buchsta-

benorientierte (zeichenorientierte). Die Eingabe von Wörtern (Befehle) über die Tastatur dauert schlicht und einfach länger als das Zeigen auf eine Menüposition. Gleichzeitig lernt man grafische Oberflächen schneller als zeichenorientierte. Denn in diesem Falle muß man sich nicht noch Schreibweise und syntaktische Regeln merken, sondern dann nur noch die Bedeutung der Befehle, das heißt was möglich ist und wo es möglich ist. Das Wo meint hier in welchem Menü und unter welchen Bedingungen.

2. Kontextabhängigkeit: Der Presentation-Manager von MS-OS/2 bietet Informationen und Bedienungsmöglichkeiten situationsabhängig (kontextabhängig). Was in einer bestimmten Situation der Bedienung nicht von Bedeutung ist oder aber überhaupt nicht möglich ist, wird auch nicht gezeigt, wodurch Wahrnehmung und Aufmerksamkeit des Bedieners nicht unnötig strapaziert werden. Tieferliegende Informationen und Befehle bleiben solange in sogenannten Drop-down-Menüs, Dialogboxen oder hinter Symbolen versteckt, bis der Bediener sie benötigt und aktualisiert.

3. Die Einheitlichkeit bei verschiedenen Betriebssystemen: Da sich der Presentation-Manager von MS-OS/2 auf dem Bildschirm auf exakt die gleiche Weise präsentiert wie Microsoft-Windows 2.0 und Microsoft-Windows 386, gibt es für den Anwender nun eine einheitliche Bedieneroberfläche für Computersysteme unter MS-DOS und unter MS-OS/2, das heißt für zwei Prozessorgenerationen - 8088/8086 und 80286. Da Microsoft die erweiterten Möglichkeiten der 80386er Prozessoren schon mit Windows 386 nutzt und MS-OS/2 in dieser Richtung ebenfalls weiter entwickelt wird, kann man davon ausgehen, daß es in Kürze eine einheitliche Bedieneroberfläche für drei Prozessorgenerationen geben wird.

4. Einheitlichkeit bei allen Anwenderprogrammen: Der Presentation-Manager ist ebenso wie die Windowsversionen nicht nur die Schnittstelle des Betriebssystems zum Bediener, sondern auch die Schnittstelle für Programmierer, auf die Grafikmöglichkeiten der in Frage stehenden Computersysteme zuzugreifen. Je mehr Programmierer die grafischen Programmierhilfen, die der Presentation-Manager bietet, nutzen, desto mehr Programme werden sich unter einheitlicher Oberfläche präsentieren. Das Konzept einer einheitlichen grafischen Programmführung verkürzt auch hier die Lernzeit des Anwenders und bewirkt, daß er in kürzerer Zeit mit seinem An-

wendungsprogramm produktiver arbeiten kann. Der Anwender wird bei einem neuen Programm vieles schon aus anderen Anwendungen kennen. Das folgende Schema verdeutlicht das Konzept. Der Presentation-Manager bildet den umfassenden Rahmen, in den über Programmiersprachen die einzelnen Anwendungsprogramme eingebunden sind. Dabei können Mission-Critical-Applikationen unter MS-OS/2 besonders geschützt werden. Mission-Critical-Applikationen nennt man solche Anwenderprogramme, die besonders kritische, für die eigene Arbeit oder einen ganzen Betrieb (über-) lebenswichtige Daten bearbeiten. Sie sind in der schematischen Grafik deshalb besonders hervorgehoben (Bild 16.4).

5. Datenaustausch zwischen Programmen auf der Applikationsebene: Ein weiterer Vorteil von MS-OS/2 in der Kombination mit dem Presentation-Manager ist die verbesserte Kommunikation zwischen einzelnen Anwendungsprogrammen. Wer schon jetzt unter Windows arbeitet, kennt den Electronic-Clip-Board. Der Anwender kann zum Beispiel ein Bild aus seinem Grafikprogramm in diesem Clipboard ablegen und aus diesem Clipboard heraus in ein Textdokument hineinkopieren, das er zum Beispiel in einem anderen Fenster laufen hat. Für den PresentationManager wird es auch die Funktion des sogenannten dynamischen Datenaustausches geben. Das ist die direkte Kommunikation zwischen verschiedenen

Anwendungsprogrammen. Realisiert ist es zum Beispiel unter Microsoft-Excel und unter Windwos-386. Dort haben Sie zum Beispiel in einem Fenster Ihre Kalkulation und in einem zweiten Fenster die grafische Darstellung der kalkulierten Werte. Sobald Sie in der Kalkulation etwas ändern, sehen Sie die Veränderungen auch sofort in der Grafik. Dynamischer Datenaustausch (DDE) ist die direkte Kopplung von zwei Programmen, das heißt Daten müssen nicht in einem Zwischenspeicher, wie zum Beispiel dem Clipboard, gelagert werden. Das folgende Schema zeigt die Software-Architektur, die dem Dynamischen Datenaustausch zugrundeliegt (Bild 16.5).

Zenith unterstützt die grafischen Oberflächen durch geeignete Hardware

Unter MS-OS/2 gibt es eine grafische Oberfläche geben, die eine komplexe Informationsverarbeitung erleichtert. Von daher gewinnt auch der grafische Teil der Hardware zunehmend an Gewicht. Hier liegen die Stärken von Zenith-Rechnern, da dieses Unternehmen einer der führenden und ältesten Fernseh-Hersteller in den USA ist.

Bedenkt man, daß wir erst vor der großen Welle von Einführungen von PC's am Arbeitsplatz stehen, dann

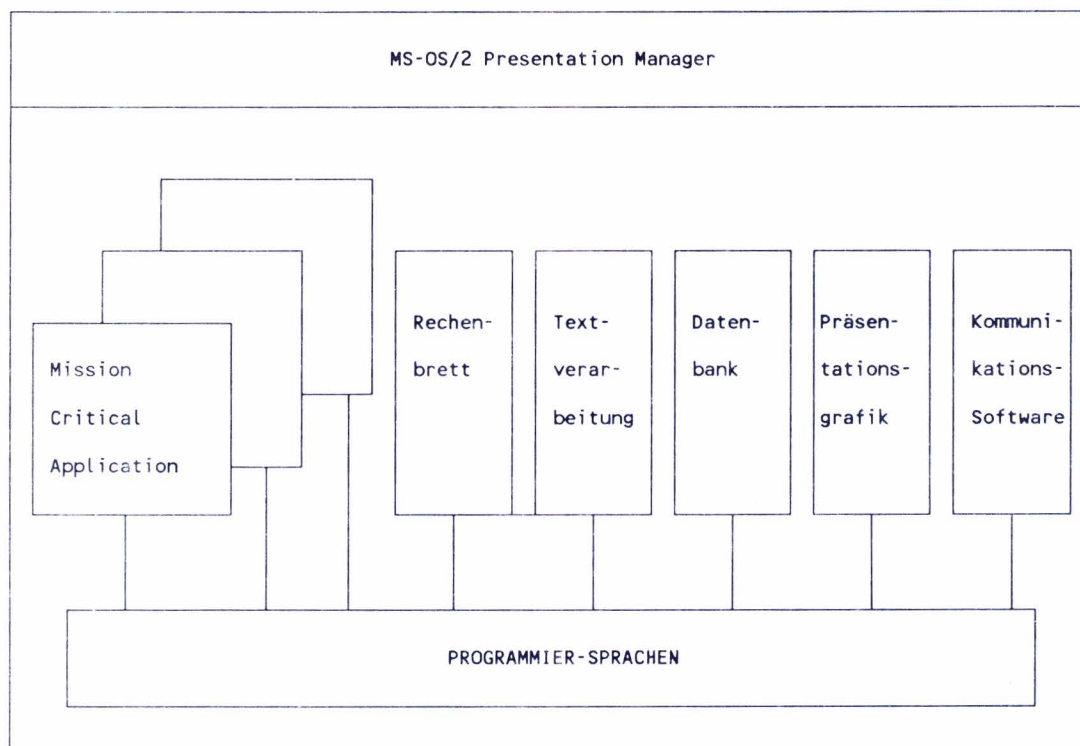


Bild 16.4

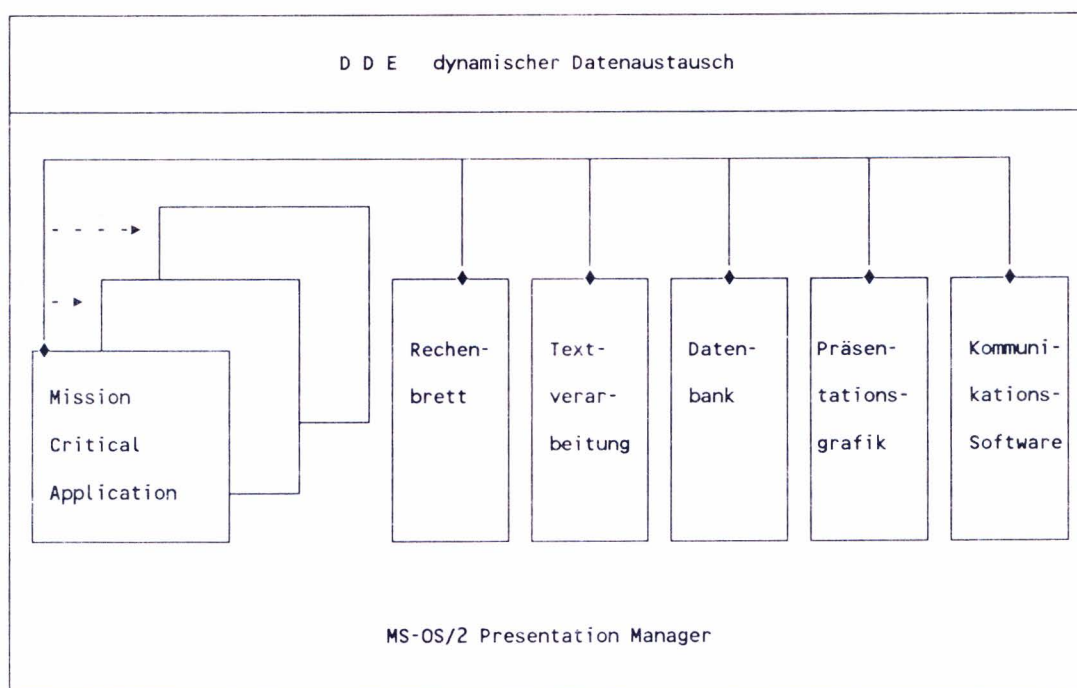


Bild 16.5

gewinnt die Frage des Displays, das heißt der Anzeige von Automatenfunktionen, zunehmend ergonomisches Gewicht. Das Problem des humanen Arbeitsplatzes hat stets zwei Aspekte: den Aspekt des rationellen Arbeitens bei rascher Auffassung und schnellem Lernfortschritt und den gesundheitlichen Aspekt der nichtschädlichen und nichtermüdenden Bildschirmarbeit. In der Mensch-Maschinen-Arbeit nimmt daher auch die Hardware-Benutzerschnittstelle, der Videoteil, eine ebenso bedeutsame Stelle ein wie die Schnelligkeit des Prozessors. Von daher scheint es mir wichtig, auf diesen Teil der Hardware in Zusammenhang mit MS-OS/2 näher einzugehen.

Zenith bietet in zwei der Desktop-Geräte, das sind der Z-286 und der Z-386, die Videokarte Z-449 an. Sie stellt die physikalische Schnittstelle zwischen der Hauptplatine und dem Monochrom- bzw. Farbbildschirm dar. Diese neue Karte sorgt für eine Bildschirmauflösung von maximal 640x480 Bildpunkten. Das ist der derzeit höchste Farbgrafikstandard (VGA) im mittleren Bereich. Für die Z-286-Modelle ist diese neue Karte Standard. Die Bildschirmkarte ist die Komponente, die neben dem Prozessor INTEL 80286 die Kompatibilität zur IBM-PC/AT- und IBM-PC/XT-Architektur herstellt. Sie wurde als separate Platine konzipiert, um auf künftige Entwicklungen auf dem Markt flexibel reagieren zu können. Diese Karte stellt zur Zeit für die Zenith-Modelle den neuesten Stand der Technik dar, da diese Karte

neben der zuvor erwähnten VGA-Auflösung (Video Graphics Array, 16 Farben, Bios-kompatibel) auch die EGA, CGA und MDA/Herkules-Auflösung liefern kann. Die MDA-Auflösung (Monochrom Display Adapter) unterstützt nicht den Grafikmodus. Dagegen ist die Herkules-Auflösung für grafische Anwendungen möglich. Sie hat einen Bildschirmspeicher von 64 kByte und stellt Grafiken mit einer Auflösung von 720x348 Bildschirmpunkten dar. Die Herkules-Karte ist wegen ihrer guten Grafikauflösung heute wohl die meistverbreiteste Videokarte im Bereich der Personal-Computer, so daß diese Auflösung als Standard-Grafikauflösung bezeichnet werden kann. Die CGA-Karte (Color Graphics Adapter) stellt den ursprünglichen einfachen Farbgrafikstandard von IBM dar. Sie unterstützt vier mögliche Farben. Die Bildschirmauflösung für grafische Anwendungen beträgt 320x200 Bildschirmpunkte. Sie ist damit weit weniger genau als die Herkules-Auflösung. Aber diese Schwäche wird durch den Vorteil der Auswahl von vier möglichen Farben aus einer Farbpalette von 16 Farben aufgewogen. Die CGA-Auflösung hat sich für viele technische Anwendungen, aber auch für Desktop-Publishing, das ist für die Herstellung von Druckmanuskripten, als nicht besonders geeignet, ja als nicht ausreichend erwiesen. Deshalb wurde im nächsten Schritt die EGA-Karte (Enhanced Graphics Adapter) entwickelt. Sie erlaubt eine Auflösung von 640x350 Bildschirmpunkten bei einer gleichzeitigen Darstellung von 16 Farben aus

einer Palette von 64 Farben. Damit erreicht die EGA-Karte Auflösungsweite, die in der Nähe der Herkules-Karte liegen. Die meisten EGA-Karten sind auch mit den weniger komfortablen Grafikdarstellungen kompatibel - also mit MDA, Herkules und CGA. Der OS/2-fähige Rechner Z-248 ist mit einer solchen Karte ausgerüstet, die sich automatisch auf die vom Monitor aus bestimmte Grafikauslösung einstellen kann. Der heute in diesem Bereich höchste Standard ist - wie schon erwähnt - die VGA-Karte. Hohe Auflösung hat den Vorteil, daß Verzerrungen so gut wie ausgeschaltet sind. Während im EGA-Modus die einzelnen Grafikpunkte durch kleine Rechtecke dargestellt werden, werden diese im VGA-Mode durch kleine Quadrate dargestellt. So werden Kreise auch wirklich als Kreise und nicht als leichte Ovale dargestellt. Dieser Standard wird bei den Rechnern der Z-286-Serie standardmäßig geliefert und bei der Serie Z-386 auf Wunsch eingebaut.

Daß gerade mit dem Auftreten von OS/2 das Thema Grafikdarstellung am Bildschirm dominant wird, hängt natürlich mit der vollen Ausnutzung der Prozessoren 80286 und 80386 von INTEL durch dieses Betriebssystem zusammen. Grafische Darstellungen bedürfen eines großen Speicherraumes, der nicht nur in der Hardware, sei es als Arbeitsspeicher oder als Festplattenspeicher vorhanden sein muß, sondern auch über den Prozessor und das Betriebssystem adressierbar sein muß. Diese Voraussetzungen sind nun vor allen Dingen von der Betriebssystemseite durch OS/2 mitgegeben. Es kann von daher mit einem explosionsartigen Schub in der Entwicklung grafisch arbeitender Software gerechnet werden. Dieser Entwicklung trägt Zenith mit seinem Hardwareangebot auf diesem Sektor Rechnung. Vor allen Dingen die Monitore können von ihrer Qualität her heute als führend betrachtet werden.

Der FTM-Monitor ZCM-1490 (FTM = Flat-Tension-Mask)

Der ZCM-1490 ist eine Eigenentwicklung von Zenith, an der insgesamt über 20 Jahre lang gearbeitet wurde. Dieser Entwicklung liegt ein völlig neuartiges, weitgehend patentiertes Konstruktionsprinzip zugrunde - die Flat-Tension-Mask-Technik. Bei Kathodenstrahlröhren mit Farbdarstellung ist in der Regel direkt hinter der Frontscheibe eine Lochmaske angeordnet. Die Lochmaske ist dafür verantwortlich, daß der unscharf auftreffende Elektronenstrahl auf

ein einzelnes Farbpixel begrenzt wird. Durch den ständigen Elektronenbeschuß erwärmt sich die Lochmaske und dehnt sich daher aus. Die Folge davon sind Unschärfen und Farbverschiebungen. Um eine Ausdehnung der Lochmaske durch Erwärmung zu verhindern, wird im ZCM-1490 die Lochmaske unter einer Vorspannung von ungefähr 10 Tonnen pro Quadratcentimeter in den Glastubus eingegossen. Weil der Abstand zwischen Lochmaske und Frontscheibe möglichst klein sein muß, erhält man damit automatisch eine völlig flache Vorderseite. Auf dieser Konstruktion aufsetzend können beim ZCM-1490 nun zwei Schritte unternommen werden, die bei herkömmlichen Monitoren ausgeschlossen sind: Zum einen kann die Energie des Elektronenstrahls erheblich erhöht werden, weil sich die Lochmaske trotz Erwärmung so gut wie nicht mehr verschieben kann. Andererseits kann zur (fast 100-prozentigen!) Entspiegelung des Monitors die aus der Raumfahrt-Entwicklung stammende OCLI-Folie verwendet werden. Diese Folie ist jedoch nur auf flachen Oberflächen verwendbar und deshalb für Monitore mit gewölbter Frontseite nicht geeignet. Durch diese Maßnahmen erzielt der ZCM-1490 enorm verbesserte Werte für Helligkeit (70%) und Kontrast (50%). Dazu kommen die Vorteile der fast 100-prozentigen Entspiegelung und der exakten Bildschirmgeometrie. Der ZCM-1490 ist ausschließlich auf die neue VGA-Norm hin konstruiert und daher für alle Rechner mit einer VGA-Karte geeignet - insbesondere auch für alle IBM-PS/2-Systeme.

Die LCD-Bildschirme: Das Supertwisted-Backlit-Verfahren

Die Hauptvorteile von Flüssigkristallanzeigen (LCD's) gegenüber herkömmlichen Monitoren sind mit Sicherheit die extrem platzsparende Bildschirmkonstruktion und der weitaus geringere Stromverbrauch. Deshalb setzen sich LCD-Displays in zunehmendem Maße vor allem bei tragbaren PC's durch.

Bei den Flüssigkristallen, die in LCD's verwendet werden, handelt es sich um organische Kristalle. Ihre optischen Eigenschaften lassen sich auf elektrischem Wege beeinflussen. Wenn sich die Flüssigkristalle innerhalb eines elektrischen Feldes befinden, dann richten sich die einzelnen Moleküle parallel zueinander aus. Dadurch wird die Lichtdurchlässigkeit der Flüssigkristallschicht gesteuert. Man kann sich bei diesem Vorgang die Flüssigkristallschicht als Jalousie

vorstellen, die - je nach ihrer Stellung - das Licht zurückhält oder lichtdurchlässig ist.

Das Supertwisted-Backlit-Verfahren von Zenith funktioniert so, daß sich die Moleküle innerhalb der Flüssigkristallschicht ineinander verdrehen (Supertwisted-Verfahren), sobald eine elektrische Spannung angelegt wird. Auf diese Weise wird eine Flüssigkristallzelle - je nachdem, wie sie ausgelegt ist - entweder lichtdurchlässig oder lichtundurchlässig. Die Ansteuerung der Zellen erfolgt über durchsichtige Streifenelektroden, die matrixförmig angeordnet sind. Dabei wird an die Zeilen- und die Spaltenelektrode der angesprochenen Zelle eine Wechsellspannung angelegt. Auf diese Weise wird ein elektrisches Feld in der Flüssigkristallzelle aufgebaut, das die Anordnung der Kristalle steuert.

Um die Veränderungen in der Flüssigkristallzelle sichtbar zu machen, werden Polarisationsfilter benutzt, die gleichzeitig den Kontrast (das ist das Verhältnis von der Vordergrund- zur Hintergrundhelligkeit) verstärken und damit für ein außerordentlich

klares und scharfes Bild sorgen. Kontrastverstärkend wirkt auch die von ZENITH verwendete Hintergrundbeleuchtung des Bildschirms (Backlit-Verfahren). Das LCD-Display ist so völlig unabhängig von den Lichtverhältnissen der Umgebung. Außerdem ist die Anzeige aus beliebigen Betrachtungswinkeln klar ablesbar, was normalerweise bei Flüssigkristallanzeigen nicht unbedingt der Fall ist. Alles in allem wird mit diesem Bildschirmkonzept eine augenfreundliche, ruhige Anzeige bei den Portablen geboten.

Natürlich ist das mit Flüssigkristallen erzielbare Kontrastverhältnis von 1:3 nicht so groß wie bei aktiven Anzeigen. Aber gerade im Bereich der tragbaren Personal-Computer ist die LCD-Anzeige anderen aktiven Anzeigetechnologien überlegen. Denn das LCD-Display bietet - zum Beispiel gegenüber der Plasma-Anzeige - den wesentlichen Vorteil, daß ein Rechner mit Flüssigkristallanzeige netzunabhängig betrieben werden kann. Sehr wichtig scheint uns auch, daß bei einer passiven Anzeige, wie dem LCD-Display, die Augen selbst nach mehrstündiger Arbeit am Gerät nicht ermüden.

17 Die Netzwerkfähigkeit von MS-OS/2: Horizontale und vertikale Vernetzung

Der große Vorteil von Computersystemen der mittleren Datentechnik war bislang Multitasking, die Verwaltung größerer Speicher, Programmablauf- und Speicherschutz, Prioritätssteuerung und Multiuser-Betrieb. Multiuser-Betrieb meint, daß mehrere Benutzer mit der gleichen Software über denselben Datenbeständen arbeiten. Dies ist mit dem Multitasking-Betrieb nicht zu verwechseln. Multiuser-Betrieb ist das, was MS-OS/2 - für sich allein genommen - noch fehlt. Alles andere, was wir aufgezählt haben, bringt nun auch MS-OS/2 mit. MS-OS/2 ist primär ein Multitasking-Betriebssystem und kann als Single-User-Betriebssystem von sich allein aus nicht den Multiuser-Betrieb aufnehmen.

Diese verbliebene Lücke wird ein Zusatz-Betriebssystem zu MS-OS/2 ausfüllen: der OS/2-LAN-Manager. Mit ihm zusammen erreicht MS-OS/2 auf den Prozessoren 80286 und 80386 die volle Funktionalität der mittleren Datentechnik. Dabei wird ein spezielles Konfigurationskonzept, die häufig sogenannte Netzwerk-Philosophie, verfolgt. Man nennt solche Konfi-

gurationen in Mehrplatz-Systemen (Multiuser) auch Rechner-Architekturen. Über die zwei wichtigsten Architekturen sei hier kurz berichten.

Schon vor MS-OS/2 ist kontrovers über diese beiden Multiuser-Rechner-Architekturen gestritten worden. Die Multiuser-Architektur der mittleren Datentechnik ist stets vorwiegend die folgende gewesen: ein Rechner, ein Prozessor, ein Datenspeicher - alles von hinreichend hoher Kapazität - und eine Vielzahl von Terminals, das sind Eingabe- und Ausgabegeräte, die zumeist aus Tastaturen und Bildschirmen bestehen. Ein Terminal ist ein nicht-intelligenter Rechnerarbeitsplatz. Man nennt ihn nicht-intelligent, weil er ohne den Zentralrechner keinerlei Operationen durchführen kann. Er bildet eine Arbeits-Station, über die man Daten in den zentralen Rechner eingeben kann und an der man Daten aus dem zentralen Rechner sichten kann. Aber rechnen kann die Terminal-Station selbst nicht.

Im Gegenzug zur Terminal-Philosophie entstand mit dem Aufkommen der Mikros und später der Personal-Computer das Konzept der Netzwerk-Architektur. Netzwerk-Architektur unterscheidet sich von der Terminal-Architektur grundsätzlich dadurch, daß die Arbeitsplätze mit eigenständigen und deshalb intelligenten Rechner-Stationen - heute bestehend aus Personal-Computern - ausgestattet sind. Das Netzwerk-Konzept, das gegenwärtig vornehmlich diskutiert und auch von der Gesamtstrategie von MS-OS/2 unterstützt wird, geht davon aus, daß es einen zentralen Rechner gibt, der den Gesamtdatenbestand verwaltet. Die intelligenten Stationen können auf diesen Datenbestand zugreifen, ihn unabhängig vom zentralen Rechner weiterverarbeiten, um entweder eigene Datenbestände anzulegen oder die bearbeiteten Daten zum Zentralrechner zurückzuschicken. Man nennt diesen zentralen Rechner Server, weil er die anderen bedient.

Die Bildung von Computernetzen innerhalb einer Organisationsstruktur erfährt einen immer höheren Stellenwert. Die Verwendung von sogenannten lokalen Netzwerken (Local Area Networks), also die Verbindung von gleichartigen Computern unter gleichartiger Software, ist nicht nur für den reinen Datenaustausch (File Transfer) wichtig, sondern vielmehr für den raschen Informationsaustausch einzelner Daten und die Verfügbarkeit von aktuellen Daten an allen Arbeitsplätzen. Dabei spielen vor allem zwei Punkte eine besondere Rolle: Erstens die durchgängige Konsistenz des Datenbestands, das heißt, daß alle Daten - gleichgültig auf welchen Arbeitsstationen sie sich gerade befinden - miteinander übereinstimmen und keine Widersprüche bilden; zweitens der rasche Zugriff auf Daten, die permanent auf dem neuesten Stand sind, weil dies zum Beispiel für betriebliche Entscheidungen überlebenswichtig ist. Man nennt solche Daten und die damit verbundenen Anwendungsprogramme: Critical Mission Data and Applications.

Solange es nur MS-DOS als Betriebssystem für Personal-Computern gab, war beides Konsistenz und Verfügbarkeit von Daten gar nicht oder nur sehr schwer zu realisieren. Die verschiedenen Netzwerk-Hersteller mußten mit mehr oder minder raffinierten Methoden MS-DOS zum Quasi-Multiuser-Betriebssystem umschreiben, um den Netzwerk-Betrieb zu ermöglichen.

Das hat zu den unterschiedlichsten Lösungen geführt und letztlich keinen einheitlichen Standard aufkommen lassen. Für Netzwerk-Server wurden teilweise auch Multitasking-Betriebssysteme entwickelt. Aber auch hier hat sich kein Standard durchgesetzt. Die Folge davon war, daß auf dem Markt kaum Anwendungsprogramme für den Server angeboten werden. Trotz dieser Mängel, des fehlenden Standards und des schmalen Software-Angebots, haben PC-Netzwerke sich doch auf einem breiten Markt durchsetzen können. Die Gründe liegen im günstigen Preis und in der Möglichkeit, die Arbeits-Stationen auch als Standalone-Rechner benutzen zu können und dafür eine Vielzahl von Anwendungsprogrammen zur Verfügung zu haben. Außerdem kann in Netzwerken bestehende PC-Hardware integriert werden. Das sind alles sachliche und an Kostenplanungen orientierte Gründe, die für PC-Netze sprechen. Aber vieles hat sich auch unkontrolliert historisch - einfach so - entwickelt.

Zuerst brachte der PC nur Computerfähigkeit an jeden Arbeitsplatz. Er wurde zumeist angeschafft, um isolierte arbeitsplatzspezifische Probleme zu lösen. Den Mini-Computern (mittlere Datentechnik) und den Groß-Rechnern blieben Anwendungen vorbehalten, die auf verteilten Informationen beruhten, globale Funktionen für die Unternehmen hatten und deshalb von höherer Bedeutung waren. Mit dem immer höheren Einsatz von PC's entstanden im Laufe der Zeit immer mehr kleine isolierte Aufgabenlösungen, die es wert waren, mit anderen Arbeitsplätzen ausgetauscht zu werden. Vor allem im Bereich der Aufbereitung und Präsentation (Berichte und Grafiken) vorhandener Daten erwies sich der PC dank der bedienerfreundlichen und funktionsreichen Software den mittleren und den großen Rechnern haushoch überlegen. So verwundert es nicht, daß immer häufiger überlegt wird, ob es nicht günstiger wäre, zentrale Daten vom Host (Großrechner/Mainframe) dezentral in Netzwerken, bestehend aus unterschiedlichen Personal-Computern, weiter zu verarbeiten.

Neben rationellen und historisch gewachsenen Gründen für Netzwerke gibt es weitere, die medialer Art sind (Neue Medien). Netzwerke sind Medien zur Kommunikation zwischen Rechnern, zum Austausch von Ergebnissen und zur gemeinsamen Nutzung von peripheren Hardware-Ressourcen. Insofern eignen sie sich besonders gut für Arbeitsgruppen mit gemeinsamen Aufgaben. Man spricht deshalb in diesem Zu-

sammenhang auch von work-group-computing. Netzwerke dienen für die Arbeitsgruppe (work group) als Instrumente zur Vereinfachung von Datentransport und zur Einsparung von Ressourcen. Zum Beispiel muß man im Netz nicht erst eine Datei auf eine Diskette kopieren, um diese zu sichern. Netzwerke bieten in der Regel einen vollautomatischen back-up. Kopierte Disketten muß man nicht persönlich von einem PC auf den anderen transportieren. Es muß nicht jeder PC mit einem eigenen Drucker - zum Beispiel Laser, Plotter usw. - ausgerüstet werden, die für den Einzelplatz nie voll ausgelastet sind. Über den Netzwerk-Server können mehrere PC's auf einen Bestand von peripheren Geräten zugreifen.

Wie sieht nun die Architektur eines Netzwerkes aus? Sicherlich ist die Art des Netzwerkes stark abhängig von der Art der Hardware und - bisher zumindest - auch von der Software, die den Netzbetrieb steuert oder im Netz laufen soll. Es gibt Netzwerke, in denen alle Stationen prinzipiell gleichberechtigt sind und in denen jede Station Server in Bezug auf unterschiedliche Aufgaben sein kann: die eine Station Datenbank-Server, eine andere File-Server und wieder eine andere kann weitere Server-Dienste übernehmen. In solchen Netzen gibt es Probleme mit dem Speicherschutz und der Datensicherheit, denn es ist klar, daß das gesamte Netz von jeder (gleichberechtigten) Station hoch- und runtergefahren werden kann. Ob vorsätzlich oder fahrlässig - der Datenbestand kann zerstört werden. Der Schutz der gemeinsamen Daten und Programme kann nach dem gegenwärtigen Stand der Technologie am besten im folgenden Konzept gewährleistet werden, wie es auch von MS-OS/2 unterstützt wird.

Im Zentrum des Netzwerkes steht ein Server. Die Existenz dieses Rechners, dessen Prozessorleistung ausschließlich für die Verwaltung des Netzes eingesetzt wird, macht das Typische an diesem Netzwerk-Konzept aus. Die Server-Station kontrolliert erstens die Kommunikation in und mit den einzelnen Arbeitsstationen (work stations). Die zweite Aufgabe des Servers ist die Bedienung aller Teilnehmer mit Massenspeicherkapazität. Server-Stationen sind meist mit relativ großen Festplatten ausgestattet, die alle Netzwerkprogramme und alle gemeinsamen, zentralen Daten gespeichert halten. Die einzelnen Stationen brauchen deshalb auch nicht mit übergroßen Platten oder Diskettenlaufwerken ausgestattet zu sein. Nur die Anwenderprogramme laufen dezentral in den Arbeits-Stationen und müssen dort gespeichert

werden. Aber auch das ist nur für Stand-alone Anwendungen zwingend. Eine dritte Aufgabe der Server-Station besteht darin, alle an parallele oder serielle Schnittstellen anschließbare Peripherie zu verwalten und zur Verfügung zu stellen, wie zum Beispiel Drucker, Plotter und anderes mehr. Die Kapazität solcher Geräte kann natürlich in Netzwerken besser ausgelastet werden.

Um diese drei Aufgaben zu lösen, mußte in der Vergangenheit, vornehmlich im Hinblick auf das Übertragungsmedium, ein Standard geschaffen werden. Das ist mit dem Ethernet weitgehend gelungen. Ethernet ist das weitverbreitetste local area network (LAN) geworden. Was allerdings bisher nicht gelungen war, ist dies: ein Standardnetzwerk-Betriebssystem zu schaffen. Mit MS-OS/2 ist nun ein Betriebssystem erschienen, das auf den Personal-Computern beruht und Multitasking-Fähigkeiten besitzt. Es lag also nahe, unter diesem Betriebssystem MS-OS/2 nun auch ein Netzwerk-Betriebssystem zu konzipieren. Microsoft hat in enger Zusammenarbeit mit 3COM, einem der größten Hersteller von Netzwerken, das Netzwerk-Betriebssystem, MS-OS/2-LAN-Manager entwickelt, das auf MS-OS/2 aufbaut. Mit ihm werden verteilte Anwendungen, das heißt die gemeinsame Nutzung von Computer-Kapazität verschiedener Computer, in einem Netzwerk möglich.

Der von Microsoft und Ashton Tate schon angekündigte SQL-Server ist zum Beispiel eine Netzwerk-System-Software für verteilte Datenbank Anwendungen. Als offene Plattform bietet er im Prinzip jeder Anwendung seine Dienste an. Sie beruhen auf der Standard Datenbanksprache SQL (Structured Query Language). Das macht den SQL-Server zu einem echten Datenbank-Server. Er antwortet auf SQL-Kommandos, indem er Operationen des Datenbank-Managements-Systems auslöst. Speicherung und Suche nach bestimmten Daten sind dabei eingeschlossen. Eine Anwendung könnte zum Beispiel nach Adressen im Postleitzahlbereich 8000 fragen. Der Server gibt dann nur die passenden Datensätze zurück. Dies unterscheidet den SQL-Server von einem gewöhnlichen File-Server, der auf Standard-Betriebssystem-Operationen damit antwortet, daß er eine komplette Datei an die Arbeits-Station verschickt. Die Anfragen der Station muß dann eine Anwendung enthalten, die es ermöglicht, die gesuchten Datensätze zu finden. Diese Methode führt zu unnützem Datentransport und kostet in der Regel auch mehr Zeit. Dies wird von dem SQL-Server vermie-

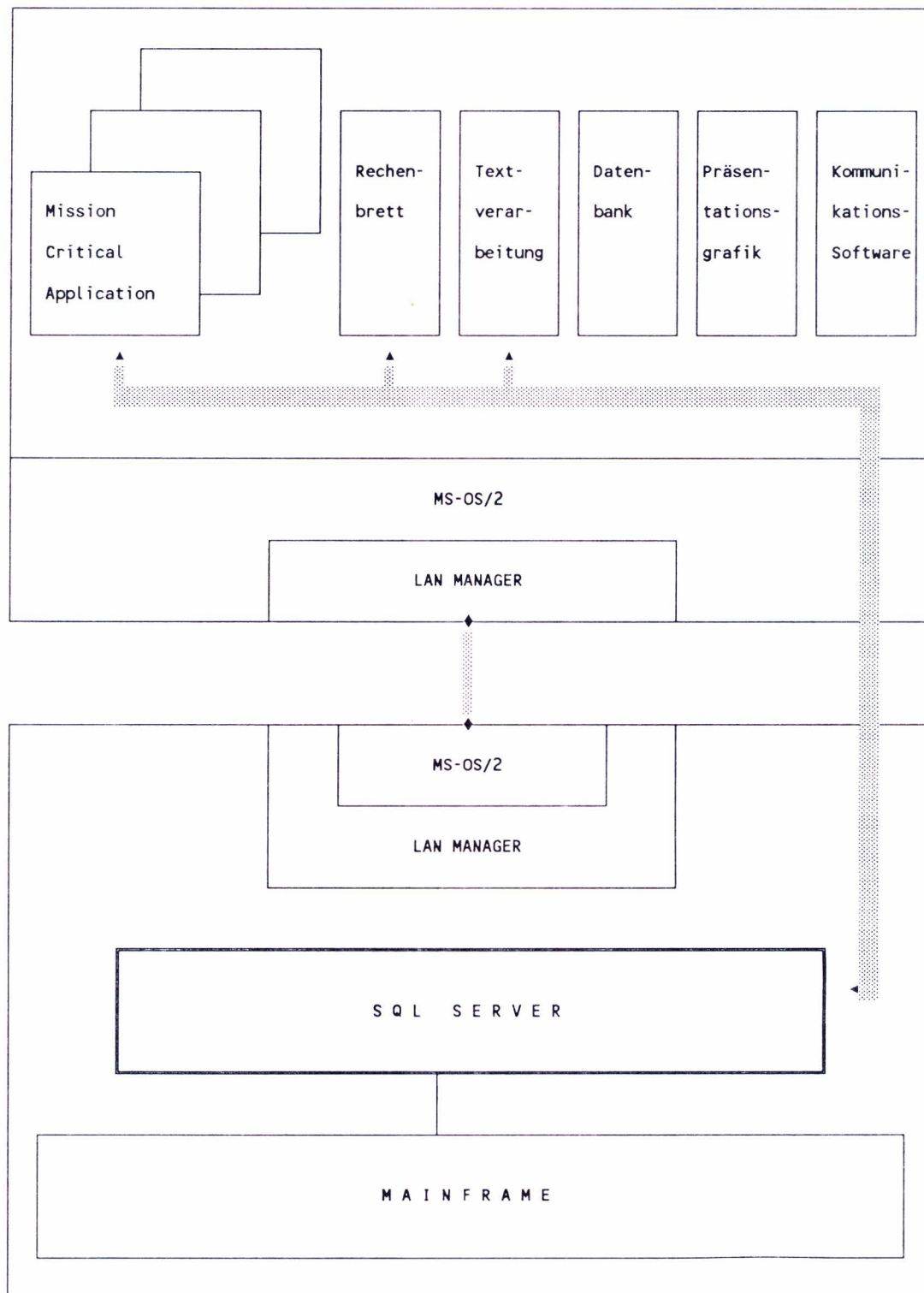


Bild 17.1

den. Außerdem lassen sich so Datenintegrität, -Sicherheit und Konsistenz besser gewährleisten (Bild 17.1).

Das Bild 17.1 zeigt eine mögliche Netzwerk-Architektur, in der der SQL-Server eingebettet ist. Da der LAN-Manager auch die Verbindung zur Mainframe

unterstützt, zeigt das Schema sowohl die horizontale Vernetzung zu einzelnen Anwendungen in verschiedenen Netzwerkstationen als auch die vertikale Vernetzung zum Host (Großrechner). Man nennt diese Vernetzung deshalb vertikal, weil sie in der Rechnerkapazität und Funktionalität aufsteigend von Arbeits-Station über Server zum Großrechner geht.

Mit einer derartigen Architektur können dann die klassischen betrieblichen Probleme, die bei der Arbeit mit zentralen Unternehmensdaten entstehen, gelöst werden. Bei diesem Konzept verliert die mittlere Datentechnik (Minicomputer) ihre Funktion. An ihrer Stelle kann ein gut ausgestatteter MS-OS/2-Rechner auf der Basis des INTEL-Prozessors 80386 treten.

Mit MS-OS/2 und dem LAN-Manager ist nun das gleiche Betriebssystem im Server wie in allen Workstations vorhanden. Dies war bisher nicht der Fall. Für Anwendungsprogrammierer eröffnet dies die Möglichkeit über ein einziges API (Application Program Interface) Anwendungsprogramme zu schreiben, die sowohl im Netzwerk-Server als auch in den einzelnen Arbeitsstationen ablaufen können. Für die Netzwerk-Server werden sich wohl Personal-Computer mit dem 80386-Prozessor durchsetzen. Unter MS-OS/2 stellen sie dem Entwickler für Netzwerk-Software 16 Megabyte Arbeitsspeicher zur Verfügung und in Zukunft 16 Gigabyte auf Festplatten.

Aber nicht nur im Rahmen großer Speicher werden Netzwerke von MS-OS/2 unterstützt, sondern vor allen Dingen mit der Möglichkeit zur Interprozeß-Kommunikation, die ja gerade im Zusammenhang mit verteilten Arbeiten von größter Bedeutung sind. Das wichtigste Kommunikationsmittel zwischen Prozessen sind die OS/2-Pipes. Pipes sind mit Briefkästen vergleichbar, in denen Nachrichten abgelegt und von anderen Prozessen aufgenommen werden können. Der MS-OS/2-LAN-Manager stellt den Entwicklern für Anwendungs-Software auf Netzwerken die sogenannten Named-Pipes zur Verfügung. Während die normalen Pipes innerhalb von Anwen-

dungsprogrammen Hilfsmittel bei der Verbindung zwischen Hauptspeicher, Bildschirm, Festplatte und Tastatur darstellen, können Named-Pipes sogar Verbindungen innerhalb eines Netzwerkes zu Bildschirmen anderer Stationen herstellen.

Der MS-OS/2-LAN-Manager basiert auf dem Microsoft-Netzwerkprogramm "Networks" und wurde für den Protected-Mode entwickelt. Er ist kompatibel zu der bereits existierenden Netzwerk-Software MS-Net und XENIX-Net. Ein mit dem LAN-Manager ausgestatteter Computer kann ohne Probleme in ein existierendes Netz, das auf MS-Net basiert, eingefügt werden - und zwar sowohl als Server wie auch als einfache Arbeits-Station.

Mit dem MS-OS/2-LAN-Manager und dem IBM-OS/2-LAN-Manager wird sinnvollerweise auch im Netzwerkbereich ein Standard geschaffen, an dem sich System- und Software-Entwickler orientieren können. Microsoft hat in den MS-OS/2-LAN Manager wichtige Netzwerkfunktionen implementiert, die den System-Programmierer unterstützen. Der LAN Manager nimmt dem Systementwickler und Programmierer die zeitraubende konzeptionelle Arbeit und das Programmieren der Systemfunktionen weitgehend ab. Der Programmierer kann sich nun ausschließlich auf die Erstellung der Anwendungssoftware konzentrieren. Die Möglichkeiten des Multitasking werden von dem LAN Manager voll genutzt, so daß auch komplexe Serverfunktionen implementiert werden können. Die verschiedenen User an den einzelnen Stationen können ihre eigene Task erhalten, die vom Server als thread behandelt und kontrolliert wird.

18 Mit MS-OS/2 in die Zukunft

In der Welt der Personal-Computer - und das ist die Büro-Welt - vollzieht sich derzeit eine der schnellsten technischen Revolutionen, die es in der Geschichte von Technik und Technologie jemals gegeben hat. Über die kurze, aber aufregende Geschichte der Personal-Computer ist an anderer Stelle berichtet worden.

Heute ist der PC in vielen Büros zum voll akzeptierten, unentbehrlichen Arbeitsmittel geworden. Und dort, wo dies noch nicht geschehen ist, steht sein Einsatz vor der Tür. Schon heute zeigt sich, daß der

Personal-Computer ein universelles Werkzeug der Büroarbeit ist. Universell heißt hier, daß er überall und für alle Aufgaben eingesetzt werden kann. Als Standalone-Gerät und Ein-Platz-System integriert der Personal-Computer vielfach schon unter der einheitlichen Oberfläche von Microsoft Windows Textverarbeitung, Tabellenkalkulation, Datenbank, Geschäftsgrafik und Projektplanungs-Software (Excel, Pagemaker, Omnis-Quarz etc.). Speziell die Integration in einem Werkzeug hat die Qualität des Arbeitsproduktes enorm gesteigert. Dieser Umstand, aber auch die allgemeinen Bedürfnisse in Betrieben,

wecken den Wunsch, die Arbeitsprodukte auszutauschen bzw. im Team leistungssteigernd an ein und demselben Arbeitsprodukt zu arbeiten.

Die technologische Entwicklung auf der Hardware-Seite hat den Personal-Computer in den Leistungsbe- reich gebracht, in dem dies möglich erscheint. Dafür sind die verbesserten Möglichkeiten der Prozessoren 80286 und 80386, aber auch die Leistungssteigerun- gen im Speicherbereich verantwortlich. Nun ist auf der Betriebssystemseite das "Missing-Link" hinzuge- kommen: MS-OS/2. Es tritt den Weg an, Standardbe- triebssystem für den Personal-Computer zu werden. Ein Standardbetriebssystem, das die erweiterten Mög- lichkeiten, die in der Hardware aufgetreten sind, voll ausschöpft.

Damit ist der Weg freigegeben für eine neue Entwick- lung von Büro-Software und für die Veränderung der Arbeitsweisen in der gesamten Büro-Welt. Anvisiert ist das Ziel, ein Netzwerk von Personal-Computern für die Büro-Arbeitsgruppe zu schaffen. Der Perso- nal-Computer wird damit vom Standalone-Rechner zum Arbeitsgruppenrechner. Das Zauberwort der nächsten fünf Jahre wird sein: Kommunikation in ver- netzten Rechnersystemen. Datenkommunikation, Electronic Mail, Telefondienste, Fax und DTP sind Hauptaufgaben, die in diesen Netzen zu verrichten sind. Damit kommt der Personal-Computer in Berei- che von Instrumenten, die strategische Anwendungen des Unternehmens betreffen. Als Beispiele wären hier zu nennen: Buchungssysteme, unternehmensinter- ne Informations-, Führungs- und Entscheidungssyste- me.

Aber auch für den Einzelplatz ist damit zu rechnen, daß MS-OS/2 unentbehrliches Betriebssystem für den PC-Arbeitsplatz wird. Über die Möglichkeiten der Prozeßkommunikation wird der dynamische Daten- austausch zwischen den spezialisierten Software- Paketen möglich. Der Freiberufliche, ebenso wie der Kleinstbetrieb muß integriert arbeiten, ohne daß der Computer bei der einzelnen Aufgabe für alle weite- ren Aufgaben blockiert ist: der typische Fall für Mul- titasking mit MS-OS/2. Für integriertes Arbeiten am Einzelplatz ist es ebenso wichtig wie für das Arbeiten in einem PC-Netzwerk, daß gewisse Software-Stan- dards durchgehalten werden. Ohne solche Standards ist weder intramaschinelle noch intermaschinelle Kommunikation möglich. Wir können davon ausge- hen, daß die nächsten fünf Jahre der Entwicklung im

Zeichen solcher Standards stehen. Die wichtigsten Standards zeichnen sich schon ab:

1. Die grafische Oberfläche des Betriebssystems als die Anwender-Schnittstelle. Das wird Microsoft- Windows bei den MS-DOS-PC's sein, und für MS- OS/2 ist dies die Weiterentwicklung von Windows, nämlich der MS-OS/2 Presentation Manager.
2. Die Standard-Schnittstelle zu Datenbanken im Großrechner oder in einem Rechner der mittleren Datentechnik (Abteilungsrechner) wird die Daten- banksprache SQL sein. SQL ist ja zur Zeit schon die Standard-Datenbanksprache auf Großrechnern und Minicomputern. Darüber hinaus wird mit dem Micro- soft-SQL-Server auch der MS-OS/2-PC selber zur Datenbankmaschine im PC-Netzwerk, der leistungsfä- hige Datenbankanwendungen ermöglicht, die teure Computersysteme auf Mini- oder Mainframe-Ebene mindestens ergänzen, und in vielen Fällen sogar erset- zen können.
3. Die Schnittstelle für die Kommunikation zwischen Rechnern in einer Arbeitsgruppe wird generell der MS-OS/2-LAN Manager sein. Netzwerksysteme auf der Basis von MS-OS/2-LAN Manager ermöglichen aber auch die Realisierung von verteilt arbeitenden Programmsystemen, und sie sind auch die Kommuni- kationsschiene, die jeden PC im Unternehmen über geeignete Gateway-Funktionen im Server-PC mit übergeordneten Rechner- und Kommunikationssyste- men verbindet.
4. Last not least bietet MS-OS/2 die Bedingungen für Datenerfassung, Verarbeitung und Datenausgabe im On-Line-Multitasking-Verfahren.

Nur wenn solche Standards gelegt sind und gepflegt werden, ist die für jeden Rechnerbetrieb lebenswichti- ge Qualität und Unterstützung auf Dauer kostengün- stig zu gewährleisten. In Zusammenhang mit diesen zukünftigen Standards werden die bereits heute instal- lierten 20 Millionen MS-DOS-Rechnern nicht abge- hängt, sondern sie können integriert werden und am Datenaustausch teilhaben.

Auch im Hinblick auf dieses Problem sind gewisse Standards Grundvoraussetzung. Windows als Benutz- eroberfläche wird es möglich machen, bei geringfügi- gen Änderungen Software der neuen Generation auf die Rechner der alten zu übertragen - und umge- kehrt. Windows als graphischer Standard wird es

möglich machen, die Software der Zukunft portabel zu halten. Die zweite Komponente, die diese Portabilität garantiert, ist das API (Application Program Interface). Zwar sind auch hier einige Grenzen zu beachten, die für MS-OS/2 und MS-DOS durch die FAPI (Family Application Program Interface) umrissen sind. Aber mit diesen Beschränkungen kann man als Programmierer leben, beim gleichzeitigen Gewinn von Standardisierung und Portabilität. Darüber hinaus bieten die Entwicklungswerkzeuge von Microsoft, also zum Beispiel Microsoft-C und MASM, der Macroassembler, Unterstützung für die Entwicklung von Programmen, die auf MS-DOS und MS-OS/2 ablaufen können.

Die PC-Welt bleibt auf diese Weise in sich konsistent. Für viele Benutzer bleiben die Unterschiede sogar unbemerkt. Wie wir gesehen haben, ist die Syntax der meisten Befehle für MS-DOS und MS-OS/2 dieselbe. Windows wird für beide Betriebssysteme die einheitliche Benutzeroberfläche bilden. Das garantiert identische Bedienung für die alte und die neue Welt. Dasselbe gilt für die Integrationseigen-

schaften, die unter Windows für beide Betriebssysteme gleich sind. Der einzige Unterschied besteht in der Performance, das heißt in der inneren Auslastung der Kapazität der Prozessoren. Die sehr ähnlichen API's garantieren, daß der Mehraufwand für Software-Entwicklung gering ist. Aber der Vorteil der Nutzbarkeit auf PC's unterschiedlicher Generationen, ist groß. Wenn sich diese Standards durchsetzen, dann stellt sich die Büro-Welt von morgen in der folgenden Architektur dar: (Bild 18.1)

Die Horizontale dieser Graphik gibt die Steigerung in der Hardware-Technologie der Prozessorgenerationen wieder, die Vertikale spiegelt den Zuwachs an Funktionalität der Betriebssysteme, das heißt den Zuwachs an Möglichkeiten, komplexe Aufgaben zu verrichten.

Die Problematik der Vernetzung von Personal-Computern ist schon lange ein wichtiges Thema, aber erst jetzt, wo MS-OS/2 die Hardware-Möglichkeiten für die ganze Breite der Anwender zugänglich macht, kommt dieses Thema in den Bereich standardisierter

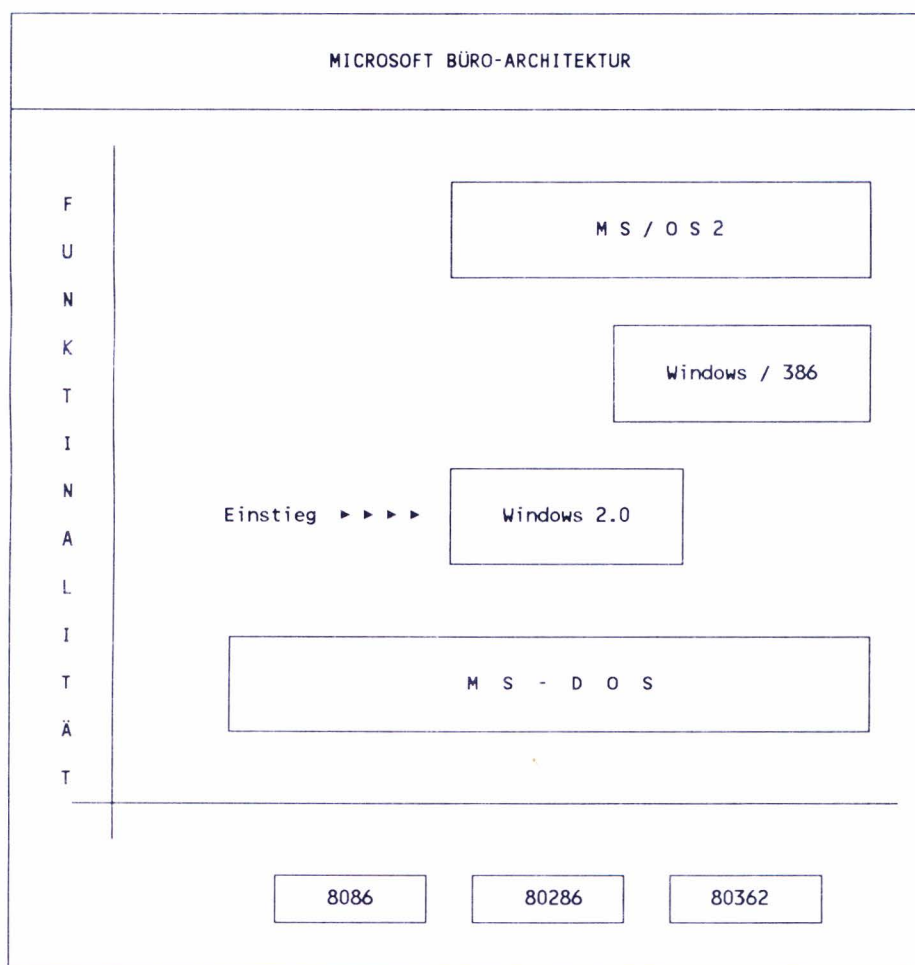


Bild 18.1

Verwirklichung. Dies gilt nicht nur für die Vernetzung von PC's in Betrieben, sondern auch für die Vernetzung von Einzelplatzsystemen in öffentliche Netze. MS-OS/2 ist zwar kein Multiuser-Betriebssystem im traditionellen Sinne, sondern im Grundkonzept ein Single-User/Multitasking-Betriebssystem. Aber parallel zur Entwicklung dieses Multitasking-Systems verläuft die Entwicklung des MS-OS/2-LAN Managers (Local Areal Network Manager für MS-OS/2). Der MS-OS/2-LAN Manager realisiert das sogenannte Client-Server-Konzept. Das heißt, von einem PC, der die Server-Funktion ausübt, werden die Klienten- oder Arbeitsplatz-PC's mit den Daten bedient, die sie brauchen. Dieses Konzept unterscheidet sich vom klassischen File-Transfer-Konzept dadurch, daß nicht ganze Dateien von der Zentrale in die einzelnen Endstationen-Rechner verschickt werden, sondern speziell jeweils die einzelnen Daten aus dem File, die vom Klienten von der Server-Station benötigt werden. Zur Steuerung dieser Dienste bietet der MS-OS/2-LAN Manager dem Programmierer ein leistungsfähiges API (Application Program Interface), homogene Schnittstellen für die einzelnen Stationen (u.a. die sogenannten Named Pipes, mit denen man Programm-zu-Programm Verbindungen herstellen und betreiben kann) und die hohe Flexibilität von Transportprotokollen, die auf

die jeweiligen Anforderungen angepaßt werden können (Bild 18.2).

Auch in diesem Client-Server-Konzept spielt die grafische Benutzeroberfläche von Microsoft-Windows eine wichtige Rolle. Der MS-OS/2-LAN Manager bildet die Basis und bedient sowohl MS-DOS-Maschinen als auch MS-OS/2-Maschinen. Beide sind ausgerüstet mit der Windows-Oberfläche, die ihrerseits den dynamischen Datenaustausch zwischen verschiedenen Applikationen im Rechner gewährleistet. Da diese modernen Standard-Applikationen wie Textverarbeitung, Datenbank, Grafik, Kommunikation und Tabellenkalkulation allesamt zunehmend über Makros zu steuern sind, das heißt selber Programmiersprachen ausbilden, lassen sich in ihnen komplexe Anwendungen programmieren.

Solche End-Anwendungen bilden das eigentliche Interesse des Endbenutzers. Microsoft-Excel ist das erste Standard-Paket, daß mit einem solchen Programmentwicklungssystem, auch Embedded Language genannt, ausgestattet ist. Zwischen den konkreten End-Anwendungen ist unter der Windows-Oberfläche bzw. unter dem MS-OS/2-Presentation-Manager ebenso Datenaustausch möglich wie zwischen der Server-Station, die vom MS-OS/2-LAN Manager ge-

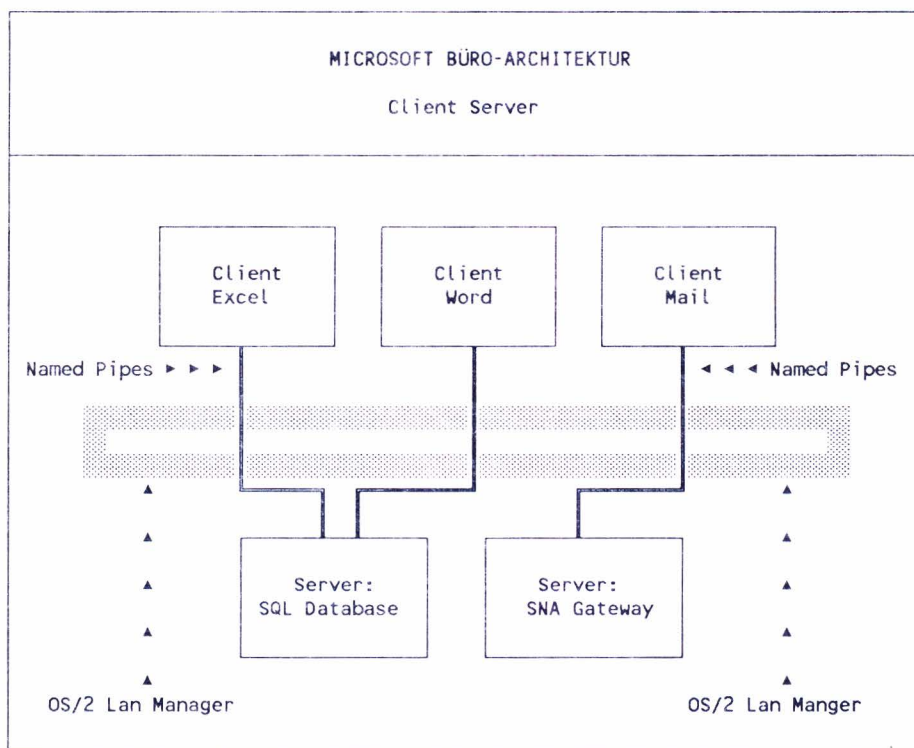


Bild 18.2

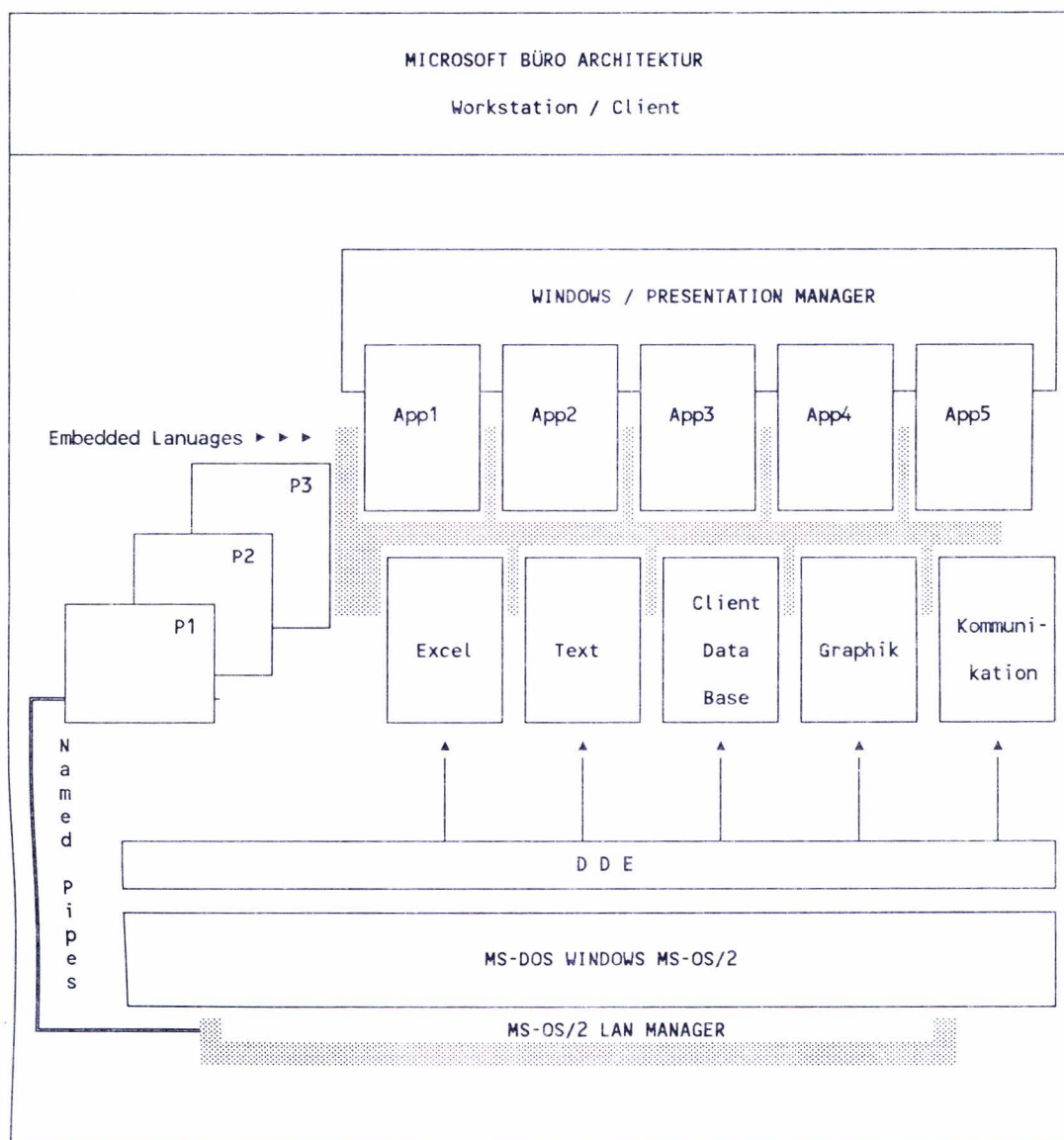


Bild 18.3

steuert wird, und dem MS-DOS- bzw. MS-OS/2-Personal-Computer. So stellt sich die Architektur zukünftiger Arbeitsplatz-PC's wie folgt dar: (Bild 18.3)

Für die Server-Dienste steht es an, eigene Software zu entwickeln. Teilweise ist sie schon angekündigt und ihre Veröffentlichung im breiten Markt steht bevor. Dies gilt etwa für den SQL-Server als Datenbank. Ebenso sind aber bereits zum Beispiel SNA-Gateways angekündigt worden. Man kann ab Mitte Mai 1988 mit vielen weiteren Produktankündigungen auch im Bereich der OSI-Gateways oder anderer Anbindungen, wie zum Beispiel Fax, Telex/Teletex usw., rechnen (Bild 18.4).

Die ersten Erfahrungen von Software-Entwicklern im Bereich des Netzwerkes sind durchgängig positiv.

Gleichzeitig bietet Microsoft Symposien und Schulungen für MS-OS/2-, Windows- Presentation Manager- und Netzwerkprogrammierung an. So kann man sagen, daß die Zukunft begonnen hat. Die ersten Schritte zur Marktdurchsetzung der neuen PC-Generation sind getan worden. Im Zentrum dieser Entwicklung steht MS-OS/2 als die Instanz, die diese Entwicklung zu allererst möglich macht.

Die Standardisierung bedarf der Unterstützung aller, das heißt der großen Software-Entwickler wie der großen Hardware-Hersteller.

Die Liste der Hersteller, die diesen Trend früh erkannt haben und ihn aktiv unterstützen, ist bereits sehr lang. Jeder, der heute einen Rechner kauft, sollte in seine Kaufentscheidung folgende Überlegun-

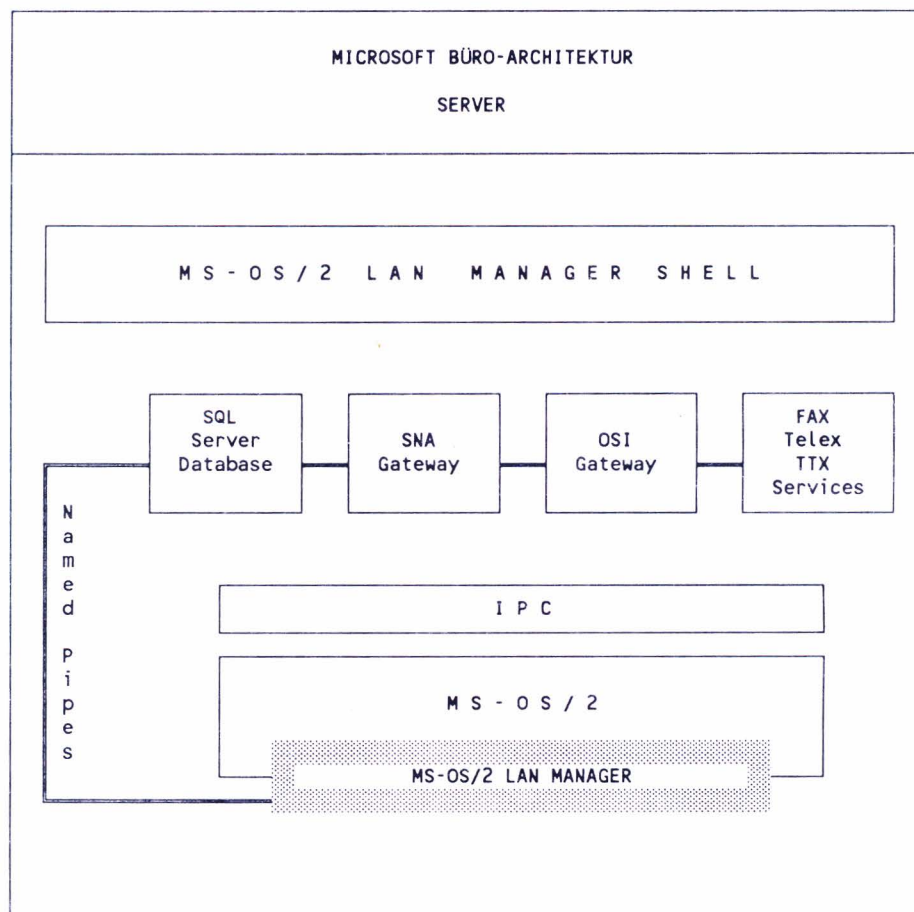
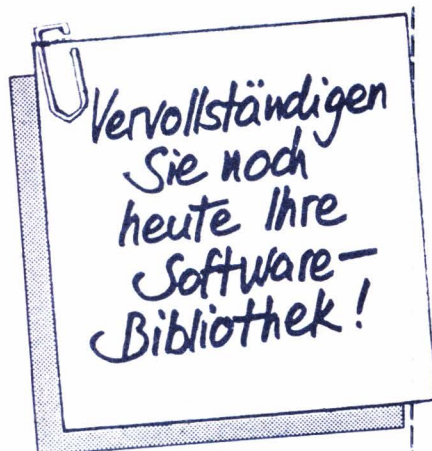


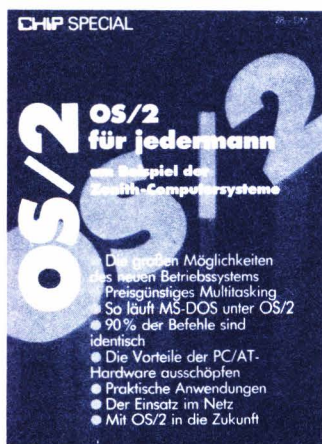
Bild 18.4

gen integrieren: Will ich an der sich abzeichnenden Entwicklung aktiv teilhaben oder genügt mir die passive Teilnahme durch den Kauf eines MS-DOS Rechners? Wer allerdings aktiv teilnehmen will, sollte bedenken, daß dies nur über den Kauf eines MS-OS/2-Rechners möglich ist, der gleichzeitig die "alte" MS-DOS Welt beinhaltet (die übrigens deswegen keineswegs "zum alten Eisen" gehört, sondern sich im Gegenteil sehr rüstig und voller Innovations-

kraft zeigt). Auch wer jetzt noch nicht im Netz arbeiten will, kann nur allzu bald auf die Notwendigkeit stoßen, sich ins öffentliche oder in ein betriebliches Netz einschalten zu wollen oder sogar zu müssen. In diesem Sinne wäre nur eine Entscheidung ratsam: der Kauf einer Hardware, die die Möglichkeiten und die Unterstützung von MS-OS/2 mindestens als Option bietet.



Bestellen Sie die Ihnen noch fehlenden CHIP-SPECIAL. Mit den nebenstehenden Karten geht's ganz leicht. Einfach ausfüllen und ab die Post!



1 Bestellkarte für weitere CHIP-SPECIALS

☐ **Ja**, senden Sie mir bitte die angekreuzten SPECIALS zu den genannten Preisen zuzüglich Versandkostenanteil DM 3,50 im Inland (Versandkostenanteil für das Ausland DM 6,-)

Ich bezahle erst, wenn ich Ihre Rechnung erhalten habe.

Anzahl	Titel	Best.-Nr.	DM/sfr.	öS
	Turbo Prolog, Ausg. 1 (inkl. Disk. 5,25")	0540	28,—	230,—
	Turbo-Basic, Ausg. 1 (inkl. Disk. 5,25")	0959	28,—	230,—
	Professionell arbeiten mit C, Ausg. 2	0964	28,—	230,—
	Turbo-Pascal 9	0972	28,—	230,—
	MS-DOS/1 (inkl. Disk. 5,25")	0949	49,—	380,—
	MS-DOS/2 (inkl. Disk. 5,25")	0951	49,—	380,—
	MS-DOS/3, Turbo-Datenbank (inkl. Disk. 5,25")	0969	49,—	380,—

Datum

Unterschrift

Bitte genaue Anschrift auf der Rückseite angeben.

2 Wünsche an die Redaktion

Am vorliegenden ERSTEN OS/2-SPECIAL gefällt mir:

gut: _____

weniger: _____

nicht: _____

Folgendes Thema sollten Sie unbedingt im nächsten Heft abhandeln:

Ich könnte folgendes selbstentworfene Programm/Thema beisteuern:

Ich möchte gerne mehr Informationen

☐ Die SPC Informationsbroschüre

Senden Sie mir Ihre ausführliche Broschüre zu, damit ich mich über alle Einzelheiten Ihrer Ausbildungsmethodik und Ihres Leistungsangebotes ein komplettes Bild machen kann.

☐ Die SPC Schulungs-Termine

Senden Sie mir Ihre aktuelle Übersicht der Termine und Programme.

☐ Ich möchte die Schulungsräume besichtigen

Ich hätte mich gerne persönlich an Ort und Stelle informiert. Bitte rufen Sie mich zwecks Terminvereinbarung an. Tel.: _____
Schulungsort: _____

☐ Software-Anpassung durch SPC

Ihr Dienstleistungsangebot zur maßgeschneiderten Anpassung von Softwareprogrammen interessiert mich. Bitte rufen Sie mich an - Tel.: _____

☐ Beratung durch SPC

Wir haben Fragen zu unserem gegenwärtigen/zukünftigen PC-System und möchten uns von jemandem beraten lassen, der völlig neutral und unabhängig ist. Rufen Sie mich bitte zurück. Tel.: _____

Name, Vorname

Straße/Postfach

PLZ/Ort

Bitte
mit 60 Pfennig
freimachen

Antwort

**Vogel-Verlag
CHIP-SPECIAL
Leser-Service 735
Postfach 6740**

D-8700 Würzburg 1

Name, Vorname

Straße/Postfach

PLZ/Ort

Bitte
mit 60 Pfennig
freimachen

Antwort

**Vogel-Verlag
Redaktion CHIP-SPECIAL
Schillerstr. 23 a**

D-8000 München 2

CHIP SPECIAL

Absender

Name:.....

Firma:.....

Funktion:.....

Anschrift:.....

.....

.....Tel.....

Postgebühr
zahlt
der
Empfänger

SPC



Software-Schulungcenter
Personal Computer GmbH

Bitte die Anschrift des nächstgelegenen
SPC Schulungcenter einsetzen.



Gegenwart



Zukunft

SPC



Wir machen Sie anwendungssicher

Software-Schulung mit Konzept und einem einzigen Ziel – Ihren erfolgreichen PC-Einsatz in der Praxis sicherzustellen.

Erfolgsgarantie durch Erfahrung. Mehr als 8.000 durchgeführte Schulungen in Deutschland, Österreich – bald auch in der Schweiz – unsere Basis für Ihren Erfolg.

Kompetenz durch Autorisierung.

Die führenden Software-Hersteller haben SPC zur Anwender-Schulung autorisiert und sorgen für Letztstand-Wissen, Erfahrungstransfer, Trainer-Fortbildung. **Praxisnahe Intensivschulung** nach dem Prinzip "learning by doing" – in lernfreundlicher Umgebung, in kleinen Gruppen – jeder an seinem PC. **Eigene Seminarunterlagen**, die

perfekte Software-Anwendung vermitteln – wertvolle Nachschlagewerke für später.

Nachbetreuung. Sollten in der Alltagspraxis Fragen entstehen – unsere Trainer geben jede Auskunft.

Dienstleistung aus unserer Praxis – für Ihre PC-Praxis. Software-Anpassung oder Systemberatung aus neutraler, unabhängiger Position.

Fragen Sie nach unseren Schulungsterminen. Senden Sie uns die nebenstehende Antwortkarte. Bitte die Anschrift des nächstgelegenen SPC Schulungszentrum einsetzen.

Oder rufen Sie uns einfach an:

Schulungszentrum Personal Computer GmbH.

D-8000 München 81, Stefan George Ring 22, Tel. 089/932 083. **A-1080 Wien**, Josefstädter Straße 74, Tel. 0222/48 58 08, 48 53 51. **A-5020 Salzburg**, Reichenhaller Straße 6-8, Tel. 0662/84 43 71.

A-8011 Graz, Sackstraße 3-5 (Hauptplatz), Tel. 0316/70 24 26. Die Niederlassung in der **Schweiz** war zum Erscheinungs-Zeitpunkt von CHIP SPECIAL in Gründung – informieren Sie sich bei einem SPC Schulungszentrum.

In dieser Ausgabe:

Autor: Dr. Norbert Meder

- Die großen Möglichkeiten des neuen Betriebssystems
- MS-OS/2 verarbeitet DOS-Dateien und Programme
- MS-OS/2 bietet ein preisgünstiges Multitasking
- Die Vorteile von MS-OS/2
an praktischen Anwenderbeispielen
- Die notwendigen Hardware-Voraussetzungen für MS-OS/2
- Die Vorteile der Hardware ausschöpfen,
aufgezeigt am Beispiel der Zenith-Computersysteme
- Wie ist MS-OS/2 auf dem Zenith-Rechner eingerichtet?

- Zehn Befehle genügen für den Anwender: Wer jetzt mit MS-OS/2 beginnt, tut sich ebenso leicht wie mit MS-DOS
- Befehlsmodifikationen im Protected-Mode und weitere zwölf Befehle für den fortgeschrittenen Benutzer
- Der MS-OS/2 Presentation-Manager:
Die grafische Bedieneroberfläche analog zu Windows
- Die Netzfähigkeit von MS-OS/2:
Horizontale und vertikale Vernetzung
- Mit MS-OS/2 in die Zukunft

- Die nachträgliche Installation von MS-OS/2
- Von MS-DOS in vier Stunden nach MS-OS/2
- Die Benutzeroberfläche:
Der Programm-Selektor und Session-Manager
- MS-DOS und MS-OS/2 sind einander sehr ähnlich:
90 % der Befehle sind identisch
- Die Unterschiede bei den Betriebssystembefehlen:
Welche neuen Befehle sind zu beachten?
- So läuft MS-DOS-Software unter MS-OS/2:
Im Real-Mode und verändert im Protected-Mode



4 003634